Fostering Computational Thinking skills in the Last Years of Primary School

Laura Freina[1], Rosa Bottino[1], Lucia Ferlino[1]
[1] *Italian National Research Council, Institute for Educational Technologies, Genova, Italy, E-mail {freina, bottino, ferlino}@itd.cnr.it*

Abstract

The introduction of Computational Thinking (CT) in Italian compulsory schools is on the way, and there is a general need for new methodologies to support teachers' work. A one-year long learning path supporting the development of CT skills in primary school students through game making activities was defined and tested in a case study with one grade 5 class. All students in the class were actively involved regardless of their personal interest, their participation was generally high, their skills increased along the project, and the main objectives were reached. Nevertheless, results suggest that a longer time span is needed for students to master deeply the new concepts and tools. Class organization proved to be crucial and this confirms the need for an adequate teachers' training before introducing CT in classroom activities. Students demonstrated to be on the right path to approach CT through game making, but they still need to be guided in taking the player's perspective and making their games easy to be understood and played. At the end of the case study, the learning path was revised and is now in use in a three-year long project in a primary school.

Keywords: *Computational Thinking, Game Making, Coding, Primary Education;*

1 Introduction

Jeannette Wing described Computational Thinking (CT) 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" [1]. The ability to think computationally is not limited to computer scientists, but rather is needed in everyday life. At present, CT has not been uniquely defined [2, 3, 4] and its assessment is an issue [5]. Different authors suggest a wide variety of skills related to CT acquisition, such as: problem-solving, examining data patterns and questioning evidence [6]; collecting, analysing and representing data, decomposing problems, using algorithms and procedures, making simulations [7], etc. Román-González, Pérez-González and Jiménez-Fernández [8] analysed several different definitions trying to determine the cognitive abilities underlying CT. They reported six different main concepts: logic, algorithms, decomposition, patterns, abstraction and evaluation, which are at the basis of several disciplines.

In Italy there has been a discussion on the introduction of computational thinking in compulsory schools within the context of the Italian National Plan for Digital Schools, promoted by the Ministry of Education as part of a general school reform (law n.107/2015) [9]. As a first step a great number of primary schools participated in limited but widespread initiatives such as "the Code weeks" where Italy, for example, is the country scoring the highest number of participants in the European Code week since its introduction [10]. Since these activities are limited in time and participation (often they are carried out as optional activities by a limited number of teachers) there has not yet been a substantial impact on schools and curricula.

The Italian Government has recently approved the introduction of CT and coding in compulsory school curricula by 2022 [11], clarifying that this will not imply the introduction of a new subject neither in primary school nor in lower secondary school. This suggests CT will be probably handled as a transversal subject in primary school and within the existing subjects of Mathematics or Technology in lower secondary school. This situation is not unique in Europe – consider, for example, France, Finland, Sweden, Norway, which have recently completed a curricula reform following this path [2, 3].

These indications open up new opportunities but also a number of key challenges that schools will have to face, including the problem to scale-up teachers' competences, the need to identify CT key concepts in order to define objectives, assessment criteria, and possible synergies with other school subject such as, for example, mathematics [4] or other STEM subjects [12], and, in general, the difficulty to accommodate a topic in already full curricula. Furthermore, the modality of introducing CT in compulsory school classes and the evaluation of its feasibility and efficacy have also to be considered.

Even though CT is a wider concept than coding, some CT skills can be developed and fostered through coding activities. Furthermore, one of the indications which derives from an analysis of the literature in the field [13], and which is present in the Italian Ministry guidelines, is that a simple and amusing way to develop CT skills is through coding in a gaming environment.

Introducing CT through game making has proved to be motivating and engaging, making learning more attractive, and effective [14-17]). Furthermore, while coding their own digital game, students develop a deeper technological fluency, which, according to Kafai, implies "not only knowing how to use new technological tools but also knowing how to make things of significance with those tools and most important, develop new ways of thinking based on use of those tools" [18].

Actually, most of the basic programming concepts are used when developing a simple digital game. According to Zaharija, Mladenović and Boljat [19], digital game making would result in:

- Understanding that computers have to be told everything they are meant to do.
- Knowing that programs execute by following precise and unambiguous instructions.
- Creating and debugging simple programs.
- Managing the abstraction needed to understand and represent the problem.
- Using logical reasoning to predict the behaviour of simple programs.
- Knowing the basic programming blocks to produce sequences of commands, loops, variables, decisions, etc.

Bermingham et al. [20] report that many studies are based on individual game making activities and peers are involved only in the evaluation phase by playing with games and providing feedback. A closer collaboration is envisaged to increase creativity, facilitate communication, as well as enhance a sense of classroom community, which encourages students to share findings, and ask and give help to peers. Furthermore, it would give students the possibility to practice their workgroup skills.

The Institute for Educational Technologies of the Italian National Research Council (ITD-CNR), has been investigating the impact of off-the-shelf digital games on the development of basic transversal skills [21] that are enacted in several primary school subjects. Initially, some studies were focused on the development of logical reasoning [22, 23]. Later, a project was organized aiming to foster primary school students' visuospatial abilities and assessing the impact of such a training on their performance in mathematics [24]. During the development of these projects, even though students enjoyed playing with most of the selected games, they showed a deeper commitment when they could be more creative and active. For example, creating virtual houses in Minecraft (a sandbox digital game where players can build objects with a variety of different blocks in a 3D virtual world) was particularly appreciated by nearly all the students [24].

Starting from this observation, and from results of previous research projects on collaborative programming with tangibles in primary schools [25], a new project was started. The main goal of the project was to define, test and refine a learning path focused on fostering some CT abilities through collaborative game making activities in the last years of primary school (students' age from 8 to 11). In the following this project and its results are briefly presented.

2 The Project

2.1 Description of the Defined Learning Path

The project was focused on the definition of a learning path aimed at fostering CT in students of the last years of primary schools. The learning path was defined having in mind students between the age of 8 and 11, but it may be of interest also for older students.

The defined learning path was organized in two phases:

- **Phase1**: a set of 10 class meetings, about one hour long, introducing some basic programming concepts and the programming environment;
- **Phase2**: a role-play, organized over 10 class meetings of about one hour and a half each, in which students played the role of game developers and created their own digital game for a client (played by a researcher).

In the design of the learning path, an iterative approach based on cycles of imagination, creation, reflections and refinement (as described by Resnick [26]), was chosen to promote autonomy, free exploration and creativity. Since no previous coding experience was required, in Phase1, each meeting focused on one specific programming concept (e.g. conditional statements, loops, variables, etc.), which was introduced with some simple examples of its application, and then free time was given to students to explore and try the new concept directly in the programming environment.

In order to support abstraction, some concrete activities were also planned to help understanding specific concepts or algorithms. Furthermore, to guarantee a full support to the students, a scaffolding structure was defined starting from a theoretical analysis of the envisaged difficulties in the game making activity [27].

In designing Phase2, a restricted set of game types was defined for students to choose from. The list of the needed competences, the specific programming skills, and the foreseen difficulties related to the proposed games constituted the basis for the definition of the scaffolding structure. A complete working sample of the proposed games and, for each game, a set of descriptions with different levels of detail were prepared. Furthermore, ready-made routines were prepared for the most difficult parts of the games (e.g. checking the distance of the actual position of a piece of a jigsaw puzzle from its final destination).

2.2 Choice of the Programming Environment

The Scratch online programming environment was chosen [28]. Scratch, created by the MIT Media Lab (http://scratch.mit.edu), is especially designed for children from the age of eight up, and has several functionalities that we considered particularly useful to reach the goals of the project. Scratch is a playful visual programming environment, where programs can be created by moving programming blocks with the mouse and snapped together like traditional building blocks. This avoids syntax errors and simplifies the coding activity. Blocks are characterized by their colour and are grouped according to their meaning. Moreover, Scratch can be accessed online, allowing students to continue their work outside school hours.

Furthermore, Scratch is a social environment, where projects can be shared and commented, it offers a wide collection of source code to copy from and stimulates ideas supporting creativity. The social aspects of Scratch are also useful to promote the idea that the developed games will be used by different players. This helped the young programmers to take the players' perspective and organize instructions and game interfaces in such a matter as to be understandable and easy to use.

2.3 The Case Study

A single exploratory case study was organized to test the learning path in a typical school environment. The case study was designed to involve the whole class, regardless of the students' personal interest in the subject. At present, most CT activities are organized outside school hours and participation is voluntary, which means that often only those students who already have an interest in computing take part to these activities.

The aim of our work was to identify the main problems and issues that can arise in a typical school environment, allowing for later adaptations of the learning path to best match it to the needs of the Italian school. The choice of testing the learning path through a case study was due to the need of evaluating it in a typical classroom, involving directly the class teacher, in the target context [29].

The case study, organized in a primary school in Genova (Italy), involved one grade 5 class (24 students, 13 boys and 11 girls, aged 10-11), none of the students had any previous coding experience. The class teacher played an active role in the whole project, and four trainee teachers from the University of Genova were involved as external observers to monitor the activities and document the project. One ITD-CNR researcher, who was present at all the meetings, supported the class teacher in organizing the activities before the meetings and offered some technical support to the students when needed.

Twenty meetings were organized, once a week, from November 2017 to April 2018. All the activities were part of the normal school curriculum and took place during school hours, either in the classroom or in the school computer lab.

As defined in the learning path, all the meetings were based on "hands on" activities. During the first 10 meetings (Phase1) Scratch was introduced and experimented through the development of small projects. In the following 10 meetings (Phase2) a role-play was organized, in which students, working in groups of four, developed a simple digital game.

Eleven different types of games were presented to the class and are available on the Scratch platform [30]. Each group could choose freely the type of game to develop:

- One group chose a maze game containing coins to be collected before reaching the exit.
- Two groups developed two versions of memory games. In these games, a set of cards, with two copies of each card, was shown upside down, and the player had to find the two cards that matched.
- Two groups developed jigsaw puzzles, in one the pieces of the original image had to be dragged to their correct location, and in the other the pieces had only to be rotated.
- One group started from the idea of clicking balloons of different colours and shapes rising in the air. The game was enriched by adding a question on a school subject every time a balloon was burst. The subject was related to the balloon's colour and shape.

All the games involved at least one "complex" issue: in the memory games students had to devise a routine to detect the correct pairs of cards; in the jigsaw puzzles they had to check the position and rotation of the pieces; in the maze, the end of the game had to be detected; in the balloon game, a list of questions had to be managed and matched with the correct answers, etc. All these concepts were successfully coded into functioning algorithms by the students.

The scaffold prepared for the learning path was used in a personalized manner: each group was given the needed support, trying, as far as possible, to push students to find their own solution to their problems. During Phase2, a specific meeting was also planned with the whole class focused on some of the most common problems that students found in developing their game. For example, during the meeting, a real life performance was organized to help students work out the correct algorithm needed in a memory game to detect when two chosen cards match. The game was played with real cards and a researcher guided students in collecting the information needed to compare the two cards. Students took notes and then, working with the whole class, managed to reproduce the algorithm in Scratch.

The games were presented at an end-of-year meeting with the families and the head of the school with great satisfaction of all the students. The six prototypes are shared to the Scratch Community and available in the Scratch online environment [31].

2.4 Collected Data

Data was collected both during the development of the project and at the end using the following tools:

2.4.1 Students' Diaries

During Phase1, after each meeting, students were asked to write a short diary of their experience. They were provided with a structured sheet organized into four sections:

- Report –to summarize what they did in the meeting,
- Like – to state what they liked best,
- Issue –to write about issues they found,
- Next Time – to say something about their plans for the following meeting.

They could also express their overall appreciation by choosing an emoticon (a smiley, a neutral or a sad face).

183 diaries were collected from the 24 students. Figure 1 shows the number of diaries collected for each student. 17 students handed in seven or more diaries, and at least one diary was collected for every student.

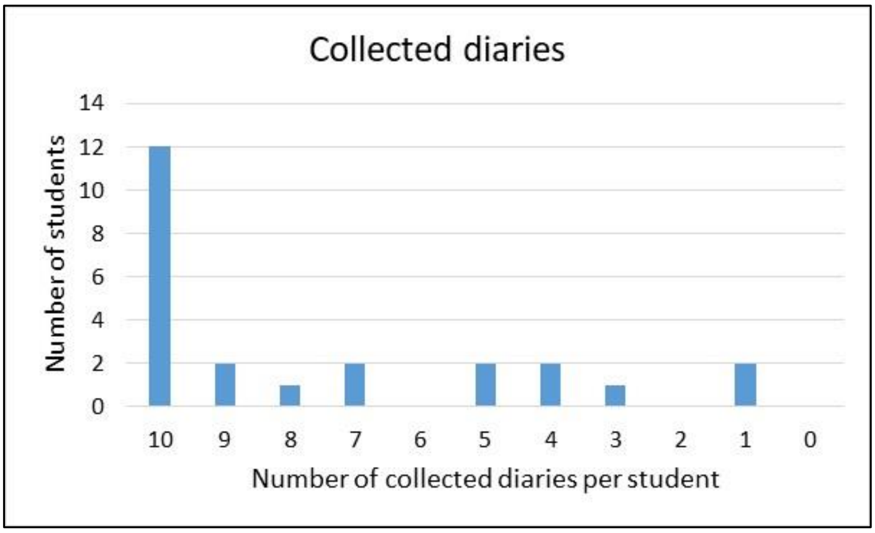


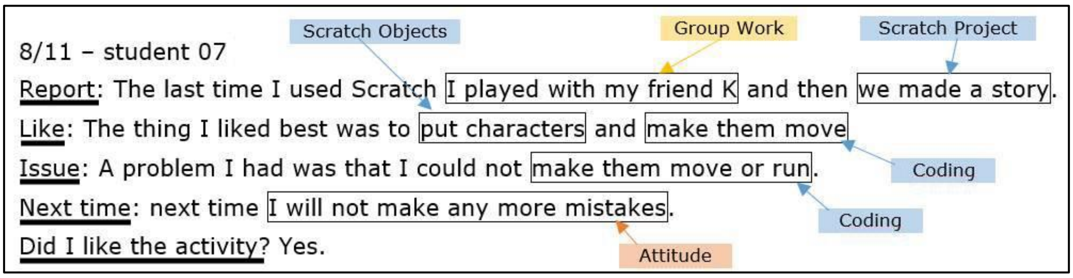
Figure 1. *Number of diaries that were collected for each student*

Texts from the 183 diaries were coded into topics by one researcher and two trainee teachers independently. Results were then merged and the identified topics were hierarchically organized into the following main categories:

- Scratch in general
  - Scratch Projects
  - Objects and Backgrounds
  - Coding
- Work management
- Social elements (related to group work, project and problem sharing)
- Attitudes (commitment, attention, effort)
- Logistics (missing loudspeakers, technical problems with computers)

- Play

Figure 2 shows an example of a diary text translated into English. Data from each section of the diary, was analysed and compared with the observation sheets collected from the trainee teachers during the class work, the final questionnaires and interviews.



8/11 – student 07

Report: The last time I used Scratch I played with my friend K and then we made a story.

Like: The thing I liked best was to put characters and make them move

Issue: A problem I had was that I could not make them move or run.

Next time: next time I will not make any more mistakes.

Did I like the activity? Yes.

Scratch Objects
Group Work
Scratch Project
Coding
Coding
Attitude

Figure 2. *An example of a diary translated into English.*

2.4.2 *Interviews with Students*

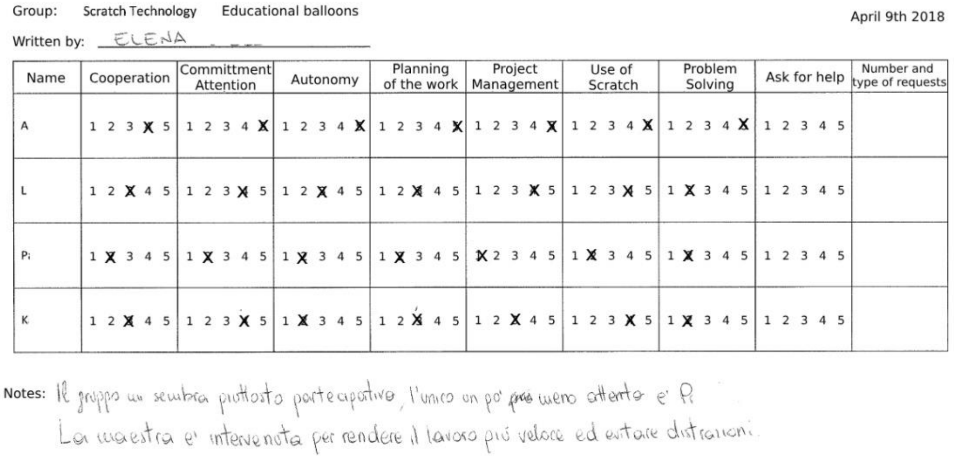
All the students were interviewed at the end of the project by the same researcher. The structured interviews aimed at verifying how much students were aware of the work done, if they remembered some problems they met and how they were solved, and if they would be able to solve those same problems autonomously. A sheet with 10 questions was used to guide the interviews, but students were free to express any observation or comment to their wish. All the interviews were recorded, typed and then analysed.

Texts from the interviews were coded into topics by one researcher and two trainee teachers independently and then, for each question, the coded answers were analysed.

2.4.3 *Observation Sheets from the Trainee Teachers*

Four trainee teachers were involved throughout the project to observe the students at work without intervening directly. Observation sheets were prepared in order to support their observations and provide a standardized tool to collect data. One researcher was also present at each meeting, supporting the technical aspects when needed and observing students and group dynamics.

One observation sheet was used at each meeting, for each group of students. The sheet was organized in eight different sections, and each member of the group was given a score for each section on a scale from 1 to 5. The observer was also free to annotate any specific remark. Figure 3 shows an example of an observation sheet.



Group: Scratch Technology Educational balloons

April 9th 2018

Written by: ELENA

Name	Cooperation	Committment Attention	Autonomy	Planning of the work	Project Management	Use of Scratch	Problem Solving	Ask for help	Number and type of requests
A	1 2 3 X 5	1 2 3 4 X	1 2 3 4 X	1 2 3 4 X	1 2 3 4 X	1 2 3 4 X	1 2 3 4 X	1 2 3 4 5	
L	1 2 X 4 5	1 2 3 X 5	1 2 X 4 5	1 2 X 4 5	1 2 3 X 5	1 2 3 X 5	1 X 3 4 5	1 2 3 4 5	
Pi	1 X 3 4 5	1 X 3 4 5	1 X 3 4 5	1 X 3 4 5	X 2 3 4 5	1 X 3 4 5	1 X 3 4 5	1 2 3 4 5	
K	1 2 X 4 5	1 2 3 X 5	1 X 3 4 5	1 2 X 4 5	1 2 X 4 5	1 2 3 X 5	1 X 3 4 5	1 2 3 4 5	

Notes: Il gruppo mi sembra piuttosto partecipativo, l'unico un po' ~~più~~ meno attento e' Pi
La maestra e' intervenuta per rendere il lavoro più veloce ed evitare distrazioni.

Figure 3. *An example of an observation sheet (translated into English).*

2.4.4 Expert Questionnaires

The class teacher, the four trainee teachers, and the two researchers who were directly involved in the project, filled in two final questionnaires. The first was centred on the project, asking for an evaluation, opinions and suggestions for future improvement. The second questionnaire was about the class. This regarded opinions and observations on the class as a whole, as well as some direct questions on those students (if any) whose attitude differed significantly from the rest of the class, both positively and negatively.

In particular, the expert questionnaire for the class had two sections devoted specifically to Phase1 and Phase2 of the project. It was asked to evaluate, in general, the class' interest in the project at the beginning and at the end of each Phase. The evaluation was on a scale of four items: very interested, somewhat interested, less interested than expected, not interested.

2.4.5 Student Questionnaires

At the end of the project, students filled in a questionnaire addressing their appreciation of the Scratch programming environment and the project activities; their evaluation of the support received, including written materials, trainee teachers, class teacher, and researchers; their evaluation of the group work and an overall score out of 10 of the project. Furthermore, some data on the students' use of internet and ICT tools were also collected to have a general overview of the class.

2.4.6 Final Scratch Test

After the end of the project, a short multiple-choice test was given to the students to assess what they learnt of the Scratch programming environment. The test was given printed on paper and students did not have the possibility to check their answer online.

2.4.7 Sociograms

Before defining the groups for the role-play (Phase2), sociograms were collected to measure the social status of the participating students. Sociograms can be a tool to assess the social peer status and hierarchy in the classroom [32], and were therefore chosen as a base, supplemented by teacher knowledge of the class, for the definition of the working groups. Data was collected both with respect to students work and play status.

The Coie and Dodge [33] method was used to determine the student's social status within the classroom: each student was assigned to one of the following categories: popular, rejected, neglected, controversial and average status. None of the students was found to be controversial. The play and the work dimensions were compared, and groups were formed starting from the most popular students. Neglected and rejected students were distributed trying to keep the groups as balanced as possible. The teacher's direct knowledge of each student also played an important role in group definition.

As a result, each group had at least one member who was considered central and responsible for the success of the work, and this demonstrated to have a positive effect on the obtained outputs allowing all groups to reach the end of the project with a fully developed prototype of their game.

2.5 Results from the Case Study

Some preliminary considerations with respect to students' work have been reported by Freina, Bottino, and Ferlino [34]. In particular:

- Students' autonomy appeared to be rather limited at the beginning of the project, when students tended to do the given work, but refrained from exploring Scratch freely. At the end of the project, data from the students' questionnaires shows that about one third of the involved students reached a greater level of autonomy in their work.
- Some students at this age still need to ground their concepts into concrete activities before moving to a virtual approach, which requires a certain level of abstraction.

Before introducing some concepts, unplugged tangible meetings proved to be useful to understand them. Furthermore, students need a long time to think about, try out and reason on the new concepts, and in those cases in which this has been possible, results were obtained.

- Scratch was chosen because it can be accessed both from school and from home, assuming that students would continue their exploration outside school hours, but this assumption was proven false. As a consequence, a much longer time period is needed to give students time to master the new concepts while incrementing their self-confidence and acquiring a deeper knowledge of the programming environment.
- The involved teachers need to have a certain knowledge of CT and coding, which goes well beyond the basic skills usually developed in teachers' training courses provided in Italian schools. Furthermore, one teacher is not enough to support the whole class. A group of CT competent teachers working together is recommended, possibly with specific knowledge of the chosen coding environment. CT activities to be proposed in class need to be planned well in advance in order to have time to discuss and try them together with colleagues, anticipate problems and think about possible solutions.

The collected data was then analysed. In the following, observations resulting from this analysis are briefly reported.

2.5.1 Students' Attitude Towards the Project

Student's attitude towards the project proved to be generally positive. In the students' questionnaire, the final average score given to the project as a whole was 8.04 out of 10. Furthermore, 54% of the students affirmed that they liked a lot programming with Scratch, and another 33% quite liked it, 13% liked it a little and nobody said not to like it at all.

The analysis of the expert questionnaires shows that at the beginning of the project students were generally interested: they liked the idea of building their own game. The class teacher and the trainee teachers agree that the class was either very (40%) or somewhat (60%) interested. During Phase1, the overall interest lowered. At the end of Phase1, the class was reported to be somewhat interested (60%) or less interested than expected (40%). According to teachers and trainee teachers, some students were disappointed because they expected to play more and program less, furthermore, due to time limits, they sometimes had difficulty in finishing the work they started. In the observation sheets, trainee teachers reported that most of the students were interested in playing with ready-made games available in Scratch, while they seldom searched for solutions to their coding issues by looking at projects shared by other people.

In Phase2, when the case study started, a similar trend can be observed: students started very (17%) or somewhat (83%) interested and finished somewhat interested (67%) or not so interested (33%). At the start of the role-play, interest raised: students started a new activity and they liked playing the role of game developers. Nevertheless, during the game development, students' interest was negatively influenced by some difficulty in solving coding issues. In some cases, the given task proved to be too complex, and the time available to acquire the different concepts was too short.

2.5.2 Group Work

In Phase1, students worked in pairs, often changing partner. Furthermore, each meeting was devoted to a different concept. Scratch projects were generally simple, requiring few programming instructions, and this allowed for enough time to share ideas and find an agreement with the partner.

Student diaries show that in Phase1, working in groups (represented by "social topics") is seldom reported as problematic. Only 6% of the social topics are in the "Issue" section of the diaries. The percentage of social topics in the "Like" section is 19%, while more than half (53%) of these topics are in the "Report" section. Apparently working in pairs was not considered by the students as a problem, but they did not particularly like it (Figure 4).

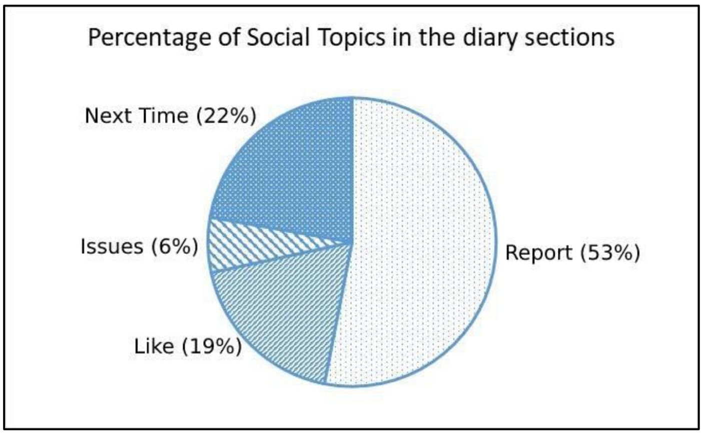


Figure 4. *Percentage of social topics in the students' diary sections.*

In Phase2, students worked in groups of four. The initial idea was that each member of the group would have a specific role in the development of their game, working cooperatively with the rest of the group, and, at the end, all the pieces would be merged in a common project. However, students did not feel confident in the coding activity and urged the support of peers, avoiding individual work.

When, in the final interviews, students were asked if working in groups was useful, 71% stated that it helped since they would not have been able to do the same work by themselves, even though 25% of the students admitted that some members of the group helped, while others did not (Figure 4). Furthermore, when students were asked if they had problems in developing the game, 44% of the reported problems were related to group work, while only 56% were technical issues on Scratch.

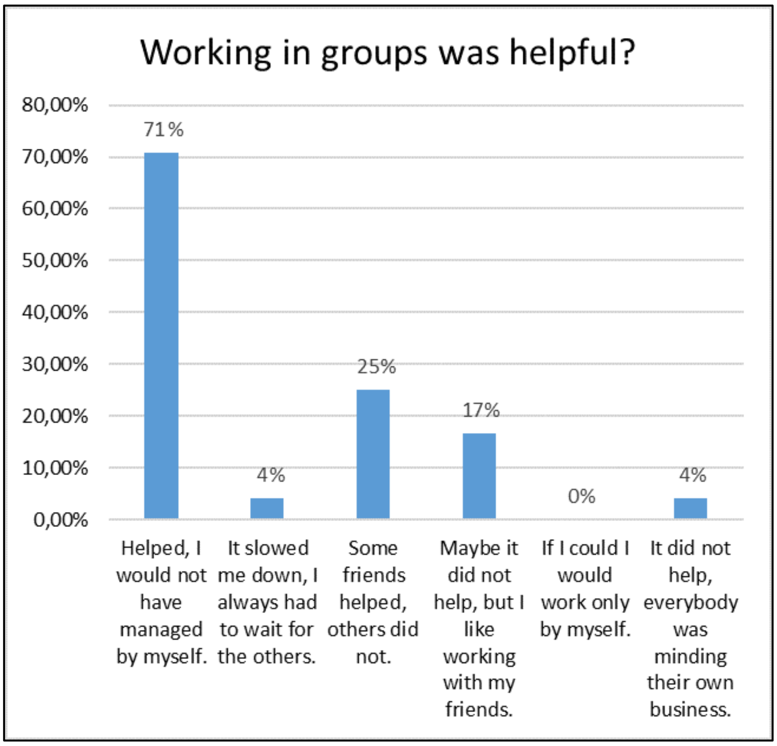


Figure 5. *Students' opinions on group work.*

Despite sociograms used for group formation proved to be generally positive, collaborative work in a group of four students caused some problems with respect to group organization, time and effort was often wasted in finding agreements and adults (teachers and researchers) often had to intervene in order to solve the situation.

The developed games were kept simple since they had to be completed in a limited time by students who had little Scratch experience. Due to the limited size of the projects, most students wanted to have a complete control over their game. They had in mind their personal game mechanics and looks, and finding an agreement with the rest of the group became difficult and frequently had to be mediated.

2.5.3 Changes in Social Status

The project showed that, in some cases, giving students a new tool to express themselves with, may offer them a new possibility to acquire a better social status within their class. This was evident in more than one case, here one situation is reported as an example.

Jane had bad scores at the sociograms, she was rejected both with respect to the work and the play dimension. She was placed in a group with other three students, among which *Paul*, who was popular in both dimensions. Soon after group work started, *Jane* and *Paul* found a good balance, they managed to work together well, discussed problems, and usually reached a common solution. The other two members of the group were not very active in coding, but they focused on the graphical aspects, searching and editing images to adapt them to their project.

Data from the observation sheets shows that *Jane* was evaluated to be high performer in all the dimensions (she scored between 4 and 5 in all dimensions but “work planning” in which she scored 3.75). *Paul* also had very similar scores, resulting as a high performer in all sections. Direct observations from the trainee teachers and, most important, from the class teacher, showed that *Jane* had a shift towards popularity in the work dimension.

2.5.4 Students’ Coding Abilities

Collected data shows that, in the process of game making, students had far fewer problems in designing the game, defining its graphics and the objects to be integrated rather than coding it.

The analysis of the data from the student questionnaires shows that most of them use internet at least once a week (88%), but none actually worked on Scratch outside school hours. This was also due to the little availability of computers (only 33% report using a computer at least once a week), while most students access the web through tablets or smart phones. At the time of the project, Scratch was not available on tablets and smart phones and this limited its use outside school and the total time spent on coding.

Analyzing the final interviews, when students were asked how they decided which elements to put in their games (objects, graphics, etc.) most of them described the decision made with the group (40%) or the fact that they looked at other examples found online (29%), and none reports asking for help. On the other hand, when it comes to solving coding issues, only about 24% affirms to have reasoned and tried, while around 30% simply asked for help and followed tutors’ instructions. The remaining 45% either did not remember or reported specific examples without stating how they were solved. It is also significant that only 6% of the students, at the end of the project, affirmed not to remember how the objects and graphics were chosen, while 27% of them did not remember how their game was coded. None reported searching Scratch on-line environment with the aim of finding parts of code.

Nevertheless, 75% of the students passed the final multiple choice test on Scratch, scoring at least 6 out of 10, and showing a good knowledge of the most common instructions. In the test, all questions were very focused: e.g. the correct sequence of blocks needed to move an object with the arrow keys had to be selected. Therefore, the test regarded a detailed knowledge of the Scratch blocks, objects, environment, etc. More than half of the class (57%) scored eight or more out of ten, and only few students demonstrated to have very little understanding of the programming language.

The analysis of the observation sheets allows for an interpretation of these data. For each section of the observation sheets, students were given an average score by considering all the data collected during the project. This score (that can range from 1 to 5) is a general evaluation of the average performance that each student showed in each section of the observation sheet. Students were then grouped in the following ranges:

- High performers: score ≥ 3.75
- Average performers: > 2.00
- Low performers: ≤ 2.00

Table 1 reports the data from the observation sheets with respect to the two dimensions of “use of Scratch” and “problem solving”. “Use of Scratch” was defined as the ability to use the chosen tool correctly to solve a defined task, while “problem solving” refers to the ability to find a solution to a more complex problem. 29% of the students can be considered high performers in the use of Scratch (7 out of 24), and 21% are high performers in problem solving (5 out of 24).

Table 1. *Percentage (to the integer) of students assigned to the different ranges of performance according to their average performance as observed by the trainee teachers.*

	Use of Scratch	***Problem solving***
High performers	29%	21%
Average performers	58%	59%
Low performers	13%	21%

There is, apparently, a contradiction between these results and the final Scratch test: the Scratch test was passed by 75% of students, while less than 30% were considered proficient in solving problems and using the tool. This is probably due to the fact that the multiple choice test requires students to recognize the correct instructions among others, which is an easier task than identifying the commands needed to solve a task and finding them in Scratch. This shows that the learning path may need to focus much more on higher problem solving abilities rather than on the specific coding activity.

2.5.5 Communication

With respect to the developed games, one element that raised to our attention, is the fact that students are often self-centred: they develop the game for themselves, they try the game and they are generally satisfied when they manage to make it work appropriately. But they seldom focus on the fact that, after the game is shared in the Scratch environment, other people, different from themselves, may wish to play it.

All students demonstrated to be pleased to share their work when they considered it complete and were proud when their peers used their game. But often they forgot to include instructions in their work making it difficult for different players to understand how to play the game.

With this respect, it is very useful to refer to Scratch as a social environment, showing students that when they play with someone else’s game, they need to understand how to interact with it. Furthermore, the official presentation of the game at the end of the year worked really well by increasing motivation and giving a deadline by which the game had to be completed. Before the final presentation, rehearsals were organized in class, and peers were very active in giving their comments and suggestions to make the presentation better.

2.5.6 The Point of View of the Teachers

The project was evaluated positively by the class teacher, who affirmed that the duration and frequency of the planned meetings were adequate, while she agreed that groups should be smaller or better arranged. With respect to project duration, the teacher affirmed that these projects should span over all the 5 years of primary school, allowing in such a way to deepen better the main concepts, increment students' autonomy and self-confidence and incrementing the integration of the coding activities with the other curricular activities.

Integration with curricular activities is considered important by the class teacher. While trainee teachers were more focused on the importance of coding and the new technologies, the class teacher was more concerned about her students developing some basic abilities, the impact of CT on the other disciplines and the possibility to use coding as a way to foster specific concepts from other disciplines.

Moreover, the class teacher and the trainee teachers noted that games could have included video interaction, music and voice recordings, original drawings from the students and possibly other media, but this was not possible due to technological limitations. Only music and recordings were used, even though limited by the availability of microphones and loudspeakers.

Finally, in order to support fully the whole class, both the class teacher and the trainee teachers confirm the need to have more than one CT competent teacher involved.

2.6 Revisions to the Learning Path

Starting from the described outcomes of the project, the learning path was adapted in order to make it more appropriate to the Italian primary school context. Another project (Project2) is currently ongoing, and it is planned to last three years, involving students from grades three to five in Scratch based activities. Younger students, from Kindergarten to grade two, are also involved in activities based on the use of tangible objects and Scratch Jr [35] (a simplified version of Scratch designed for children from 5 to 7).

The main changes that have been made are related to the following:

2.6.1 Project Duration

During the described experience, one of the main issues was the length of the learning path. Students develop their CT skills slowly and need a long time to master the main concepts. One year is not enough to learn the programming language and use it productively for an autonomous development of a simple game.

In order to address this issue, the learning path has been widened greatly, spanning over all the grades of primary school, and starting even earlier, in kindergarten. While in the early years unplugged activities and simple programming with tangible objects is mainly used, as students get older, they move on to Scratch Jr and then Scratch. Each year, activities build onto the previously acquired knowledge, allowing for a slow and deep acquisition of the coding concepts and tools.

This gives students enough time to master each concept, limiting the number of addressed new notions in each school year. When students have a longer time, they can more easily engage in finding an autonomous solution to the problems they meet, fostering their problem solving abilities and their self-confidence.

2.6.2 Autonomy

The learning path was enriched with a set of short exercises in Phase1, letting students approach each task with an incremental amount of autonomy, while gradually fading the given support [36]. Students need to progressively rely less to the close guidance from their teacher and be more active in managing their own work.

2.6.3 Play vs Coding

As reported at the final questionnaires, and directly observed during the project, students' commitment and interest lowered during the activities because they were disappointed in having to code games rather than play them. To address this issue, students may be pushed to search more for already developed examples of the game they aim at, allowing for more free play time during Phase1.

2.6.4 Group Work

When students are acquiring coding abilities, it is usually best to let them work in pairs. This is probably the best balance between peer support and individual freedom, while pair programming techniques [37] guarantee the active participation of all the involved students.

As students' work focuses on more complex projects, larger groups may be more productive, but they require specific moments in which group work is organized and divided between members. When working at the computers, groups are split in pairs, so that students never work at the same computer in more than two people.

2.6.5 Class Presentations

As previously stated, students sometimes neglect the player's perspective while developing their game. In order to help them focus on this issue, in-progress presentations were added while projects are still under development. These are moments in which, every group is asked to show their work to the rest of the class. Projects do not need to be completed, but students must be able to tell their peers what they plan to do and how their final work will look like. The whole class can then comment each game, offering advice and suggestions to make it more appealing and simpler to understand. This is a very productive moment: the authors have to prepare for a presentation while the process of coding the game is still in progress, students can see others' work and get inspiration from them, and the group of authors can check if their idea works and how to improve it. In Project2, these moments are very motivating, peers participate productively and often give very good advice for improvement.

2.6.6 Sharing and Social Aspects

Sharing and discussing students' work with their peers need to be pushed more to stimulate a reflection on the experience and generate new ideas [26] while fostering social and communication aspects. To this end, in-progress presentations to the whole class were added to the learning path and are currently giving good results. Nevertheless, students still need some guidance to open up to the whole Scratch community. A class level sharing space is recommended to promote search and discussion within a protected space. Students are asked to share their creations to let their class mates see and comment them throughout the whole project, so as not to limit peer support to the formal presentations.

3 Discussion and Conclusions

A learning path fostering CT skills through a game making activity in the last years of an Italian primary school was defined and then tested in a case study involving one grade 5 class. Unlike other experiences, students did not participate on a voluntary base, but the whole class and their teacher were involved.

Results from the case study showed some strengths and weaknesses of the learning path. Among the main strengths, the role-play was proved highly motivating. Students, working in groups, managed to develop simple, yet complete games that were presented at an end-of-year meeting with the families and the head of the school with great satisfaction of all the students. Nevertheless, during group work, in Phase2, several social conflicts were observed. In the revision of the learning path, attention was paid to the organization of group work, adapting the structure of the activities and including team work as a learning objective.

The main weakness was the limited amount of time available. The development and consolidation of CT skills spans over all the primary school years, and a one-year intervention has a limited effect. A much longer time, allows to address a limited set of concepts each year, giving students the needed time to master them thoroughly. Furthermore, students need a long time and practice before being able to take full advantage of the Scratch social environment, they tend not to share their creations, look for inspiration or search for help to overcome specific issues. Another issue is the staff-intensive support needed. One single teacher is not enough to support the work of a whole class and this, in some cases, may limit the feasibility of the approach.

The learning path was revised and is now under test in a new three-year long project (Project2). Specifically, Scratch is used starting from grade 3 and initial data show already good results, especially with respect to students' self-confidence, autonomy and coding abilities.

References

[1] J. Wing, "Research Notebook: Computational Thinking - What and Why?", The Link. Pittsburgh, PA: Carneige Mellon, 2011.

[2] S. Bocconi, A. Chioccariello, G. Dettori, A. Ferrari, K. Engelhardt, "Developing computational thinking in compulsory education - Implications for policy and practice", EUR 28295 EN, 2016. https://doi.org/10.2791/792158

[3] S. Bocconi, A. Chioccariello, J. Earp, "The Nordic approach to introducing Computational Thinking and programming in compulsory education", Report prepared for the Nordic@BETT2018 Steering Group, 2018. https://doi.org/10.17471/54007

[4] D. Hickmott, E. Prieto-Rodriguez, K. Holmes, "A Scoping Review of Studies on Computational Thinking in K-12 Mathematics Classrooms", Digit Exp Math Educ, Vol. 4, Nr. 48, pp. 48-69, 2018. https://doi.org/10.1007/s40751-017-0038-8

[5] M. Román-González, J. Moreno-León, G. Robles, "Combining Assessment Tools for a Comprehensive Evaluation of Computational Thinking Interventions", In Computational Thinking Education, pp. 79-98, Springer, Singapore, 2019.

[6] P. Charlton, R. Luckin, "Time to re-load? Computational Thinking and Computer Science in Schools", Briefing 2, 27 April, The London Knowledge Lab, 2012.

[7] S. Gretter, A. Yadav, "Computational Thinking and Media & Information Literacy: An Integrated Approach to Teaching Twenty-First Century Skills", TechTrends, pp. 1-7, 2016. https://doi.org/10.1007/s11528-016-0098-4

[8] M. Román-González, J.C. Pérez-González, C. Jiménez-Fernández, "Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test", Computers in Human Behavior, Vol. 72, pp. 678-691, 2017. https://doi.org/10.1016/j.chb.2016.08.047

[9] "PNSD: Piano Nazionale Scuola Digitale", http://www.istruzione.it/scuola_digitale/allegati/Materiali/pnsd-layout-30.10-WEB.pdf

[10] "#CODEWEEK events scoreboard": https://codeweek.eu/scoreboard?edition=2018

[11] "Atto della Camera, Mozione 1-00117", approved on March 12th, 2019, seduta n. 140.

[12] K. Tsarava, K. Moeller, N. Pinkwart, M. Butz, U. Trautwein, M. Ninaus, "Training Computational Thinking: Game-Based Unplugged and Plugged-in Activities in Primary School", in: Pivec, M., Gruendler, J. (Eds.), Proceedings of the 11th European Conference on Game Based Learning. Academic Conferences and Publishing International Limited, Graz, pp. 687-695, 2017.

[13] J.B. Kafai, "Connected Gaming: An Inclusive Perspective for Serious Gaming", The International Journal of Serious Games, Vol. 4, Nr. 3, September 2017. http://dx.doi.org/10.17083/ijsg.v4i3.174

[14] [14] A. Bauer, E. Butler, Z. Popovic, "Approaches for teaching computational thinking strategies in an educational game: A position paper", In Proceedings of the IEEE Blocks and Beyond Workshop 2015, pp. 121-123, 2015. https://doi.org/10.1109/BLOCKS.2015.7369019

[15] M. Berland, V.R. Lee, "Collaborative Strategic Board Games as a Site for Distributed Computational Thinking", International Journal of Game-Based Learning, Vol. 1, Nr. 2, 2011. https://doi.org/10.4018/ijgbl.2011040105

[16] K. Tsarava, K. Moeller, M. Ninaus, "Training Computational Thinking through board games: The case of Crabs & Turtles", International Journal of Serious Games, Vol. 5, pp. 25-44, 2018. https://doi.org/doi.org/10.17083/ijsg.v5i2.248

[17] J. Earp, F.M. Dagnino, I. Caponetto, "An Italian Pilot Experience in Game Making for Learning", In ICT in Education in Global Context, pp. 171-199, Springer, Singapore, 2016. https://doi.org/10.1007/978-981-10-0373-8_9

[18] J.B. Kafai, "Playing and making games for learning: Instructionist and constructionist perspectives for game studies", Games and culture, 1(1), pp. 36-40, 2006. https://doi.org/10.1177/1555412005281767

[19] G. Zaharija, S. Mladenović, I. Boljat, "Introducing basic programming concepts to elementary school children", Procedia-social and behavioral sciences, 106, pp. 1576-1584, 2013. https://doi.org/10.1016/j.sbspro.2013.12.178

[20] S. Bermingham, N. Charlier, F. Dagnino, J. Duggan, J. Earp, K. Kiili, E. Luts, L. Van Der Stock, N. Whitton, "Approaches to collaborative game-making for fostering 21st century skills", In European Conference on Games Based Learning, p. 45, Academic Conferences International Limited, 2013.

[21] R. Bottino, M. Ott, M. Tavella, "The impact of mind game playing on children's reasoning abilities: reflections from an experience", In Proceedings of the 2nd European Conference on Game-Based Learning (ECGBL), Barcelona, Spain, pp. 51-57, 2008.

[22] R. Bottino, M. Ott, "Mind games, reasoning skills, and the primary school curriculum: hints from a field experiment", Learning Media & Technology, vol. 31, No. 4, 359-375, 2006. https://doi.org/10.1080/17439880601022981

[23] L. Freina, R. Bottino, L. Ferlino, M. Tavella, "Training of Spatial Abilities with Digital Games: Impact on Mathematics Performance of Primary School Students", In Proceedings of the Game and Learning Alliance International Conference (GALA), December 2017, Lisbon, Portugal. https://doi.org/10.1007/978-3-319-71940-5_3

[24] L. Freina, R. Bottino, "Visuospatial Abilities Training with Digital Games in a Primary School", International Journal of Serious Games, Vol. 5, Nr. 3, pp. 23 - 35, 2018. https://doi.org/10.17083/ijsg.v5i3.240

[25] R. Bottino, A. Chioccariello, "Computational Thinking: videogames, educational robotics, and other powerful ideas to think with", KEYCIT - Key Competencies in Informatics and ICT, 184-189. University of Potsdam, Potsdam, 2014.

[26] M. Resnick, "All I really need to know (about creative thinking) I learned (by studying how children learn) in kindergarten", In Proceedings of the 6th ACM SIGCHI conference on Creativity & cognition, pp. 1-6. ACM, 2007. https://doi.org/10.1145/1254960.1254961

[27] L. Freina, R. Bottino, L. Ferlino, "Game Programming in Primary Schools: Guidelines for a Project-Based Learning Path", In Proceedings of the 12th European Conference on Games Based Learning (ECGBL), Sophia Antipolis, France, pp. 108-116, 2018. ISSN: 2049-0992, e-ISSN: 2049-100X

[28] M. Resnick, J. Maloney, A. Monroy-Hernández, N. Rusk, E. Eastmond, K. Brennan, A. Millner, E. Rosenbaum, J.S. Silver, B. Silverman, Y.B. and Kafai, "Scratch: programming for all", Communications of the ACM, Vol. 52, Nr. 11, pp. 60-67, 2009. https://doi.org/10.1145/1592761.1592779

[29] R.K. Yin, "Case study research: Design and methods", Thousands Oaks, Sage, 2003.

[30] "5A Secondo Semestre", https://scratch.mit.edu/studios/4647880/

[31] "Lavori finiti", https://scratch.mit.edu/studios/5109674/

[32] B.P. Leung, J. Silberling, "Using sociograms to identify social status in the class-room", The California School Psychologist, Vol. 11, Nr. 1, pp. 57-61, 2006. https://doi.org/10.1007/BF03341115

[33] J.D. Coie, K.A. Dodge, "Continuities and changes in children's social status: a five-year longitudinal study", Merrill-Palmer Quarterly, Vol. 29, Nr. 3, pp. 261-282, 1983.

[34] L. Freina, R. Bottino, L. Ferlino, "A Learning Path in Support of Computational Thinking in the Last Years of Primary School", In Proceedings of the Game and Learning Alliance International Conference (GALA), Palermo, Italy, December 2018. https://doi.org/10.1007/978-3-030-11548-7_2

[35] "Scratch Junior", https://www.scratchjr.org/

[36] R.D. Pea, "The Social and Technological Dimensions of Scaffolding and Related Theoretical Concepts for Learning, Education, and Human Activity", The Journal of the Learning Sciences, Vol. 13, No. 3, pp. 423-451, 2004. https://doi.org/10.1207/s15327809jls1303_6

[37] L. Williams, E. Wiebe, K. Yang, M. Ferzli, C. Miller, "In support of pair programming in the introductory computer science course", Computer Science Education, Vol. 12, Nr. 3, pp. 197-212, 2002. https://doi.org/10.1076/csed.12.3.197.8618