Experiences of the T4T group in primary schools

Fabrizio Ferrari[1], Alessandro Rabbone[2] and Sandro Ruggiero[3]

[1] Primary school De Amicis - IC Regio Parco
Torino, Italia
f.ferrari@to7120.com

[2] Primary school R. D'Azeglio – DD R. D'Azeglio
Torino, Italia
alessandro@rabbone.it

[3] Primary school N. Tommaseo – IC Tommaseo
Torino, Italia
sandro.ruggiero@gmail.com

Abstract. In this paper we describe the experiences carried out in three different primary schools during several years with pupils in their third, fourth and fifth grades (i.e. with pupils from 7 to 10 or 11 years old). The focus is on programming because we consider it a new tool for pupils to express their creativity while they are learning fundamental elements of computer science. Obviously suitable program development environments must be used, for example Scratch that is our choice for introducing programming. Our teacher experiences are focused on finding contributions to defining an interesting, affordable and sustainable school curriculum for CS in primary and lower secondary school. Such curriculum should introduce computing respecting the pedagogical achievements that have been identified by educators in the decades for the different school grades, allowing pupils to perform new kinds of activities also to the benefit of those longtime recognized achievements.

Keywords: Primary school education, computer science, Scratch

1 Introduction

We would like to draw the readers' attention to some pupils experiences about coding and computational thinking carried out in some primary schools in Turin, Italy.

We are a group of teachers interested in working on Computational Thinking and Computer Science (CS) who met during the T4T (Teachers For Teachers) seminars, organized yearly by the Informatics Department of the University of Turin. One of the aims of T4T is to introduce programming environments as new expressive tools to pupils and students of all ages and meanwhile teach them fundamental concepts of Computer Science [b].

Our teacher experiences were, and continue to be focused on, finding some interesting, affordable and sustainable suggestions for a school curriculum on CS in primary and lower secondary school. For us a proposal (and in the end a curriculum) is sustainable when it introduces new concepts but also it is respectful of the pedagogical achievements that have been identified by educators in the decades for the different school grades, allowing pupils to perform new kinds of activities also to the benefit of those longtime recognized achievements. Obviously it does not reduce to cross-disciplinary activities. Also, suggestions must be affordable for schools: this often means they should propose (almost) free activities.

In section 2 unplugged activities, yet introducing to computing, are described with their connections to the pedagogical achievements considered typical of the school-children age. In section 3 the computer based activities are described that have been experimented: they normally begin with scaffolded activities of easy-programming like using the Lightbot or Code.org and progress to use the Scratch development environment to implement stories, quiz or games. In section 4 the transition to computer programming in Scratch is presented also for its possible contributions to the linguistic abilities of the school-children particularly for their writing abilities. The conclusive section gathers some very preliminary reflections on the experiences.

2 Computer Science unplugged

Computer science unplugged activities are an important component of the experiences introducing to computing because of the obvious reason that they do not require the use of an intermediate device unknown to the school-children. Especially in the unplugged activities that require identification of their body and a android - robot, children seem to benefit greatly in terms of operational thinking. Another reason, more tangible, is that they normally are inexpensive activities and this is almost a mandatory requirement in many italian schools.

Searching on the internet and in schoolbooks for programming activities in primary schools we find three basic types of unplugged activities:

- Paper and pencil activities to move an android/robot. In the beginning commands are independent on both the person who gives the command and on the robot that has to perform the commands (typically the cardinal symbols N, E, S, W are used). Then commands are provided that require robot dependent movements such as Forward, Left, Backward, Right
- Boardgames using the same movement instruction as above, but in competition among players (i.e. Cody & Roby - http://codeweek.it/cody-roby/)
- Games on giant boards where pupils act as robot/android following school-mates commands.

We will focus on unplugged activities of the last kind. Unplugged activities are missing of an important element, present in every computer-based task: error checking. During unplugged activities you can't have an immediate feedback telling you if you are right or wrong. We remember Cédric Villani recent words: "Coding, maybe,

is the only activity where pupil can correct errors by themselves" (http://www.atelier.net/trends/articles/cedric-villani-programmation-seule-discipline-enfant-realise-auto-correction_436613).

In our experience we notice that the more pupils issuing commands identify themselves as the programmers, the more the robot/pupils follow to the letter what they have to do thus the activity turns out to be nearer to the computer based commands execution with its immediate errors visualization.

During our activities in the primary school D'Azeglio in Turin, pupils use the "natural" board shown in figure 1 that is a part of the school playground. It is made of 84 squares (about 80 cm for each side - 7 x 12) five of them occupied by a slide.

Fig. 1. The school playground

For the pupils of the 4th grade, the teacher prepared the following activities.

1. First a competitive role play as suggested by "Cody & Roby" (http://codeweek.it/cody-roby/duello/). One pupil acted as an Android/Robot and his/her mates gave her/him all the instructions in order to "catch" a second pupil acting as an enemy Android/Robot trying to enter in first pupil's home-square. Pupils giving commands used only user dependent movement instructions F, R, L, B. This first exercise was useful for pupils to acquire confidence with the board and to learn to use a finished number of unambiguous instructions to communicate with the "Android/Robot".
2. To movement instructions as in activity in point 1. other commands were added for: Repeat (n), grab, leave. The teacher prepared some sheets where the playground was drawn in every detail, see figure 2. Each sheet had growing difficulties tasks (similarly to what can be found in the Farmer activity in Code.org - https://studio.code.org/s/20-hour/stage/9/puzzle/4). On the right of

the sheet pupils could write their own code on numbered lines to reach the given target: one line, one command. For the "Repeat (n)" command pupils could draw on the left of the code a "(" sign from a line to another to precisely define which were the commands to repeat. Later on pupils shared their written code for the same task to discuss (and find) the best solution in order to solve the tasks with the less number of instructions. At the end pupils tried their code on the outdoor playground verifying their solutions.

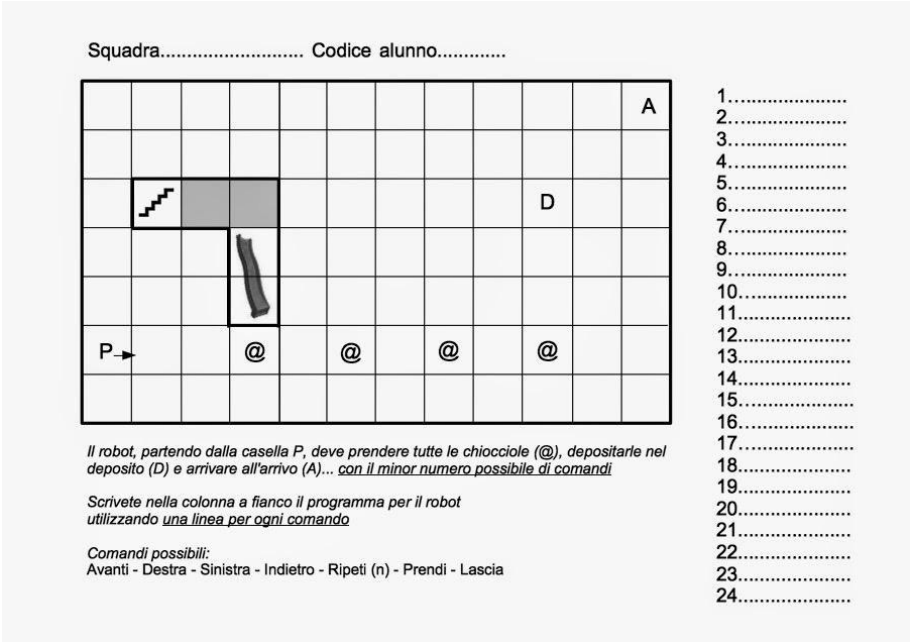


Fig. 2. The squared playground drawn on a paper

In figure 2. below the drawing of the squared playground there is the description of one of the activities suggested to the pupils: starting from P the robot must catch all the @ and bring them to point D, then the robot must go to A, with the lowest number of commands.

During unplugged activities schoolchildren are engaged in acting like they are robots moving according to a given sequence of instructions or like programmers deciding the sequence of instructions to be asked to a school-mate/robot to do a task, i.e. deciding the programs making the schoolmate/robot to do something. Soon pupils were ready to write down similar sequences of instructions making the computer to do something, that is ready to write programs for the computer with a suitable environment.

3 Computer-based programming

During the school years 2013-14 and 2014-15 the activity of computer programming (using both computer or other devices like iPad, tablet…) has been proposed in various ways and in different solutions, but all activities can briefly be classified as follows:

3.1 Puzzle solving (exercises on various websites)

We mean above all the courses offered by Code.org or by "Programma il futuro" (Italian version) in the framework of "One Hour of Code".

Still in our school labs the pupils have also experienced some puzzles with LightBot (http://lightbot.com/) and with Blockly-games' maze (https://blockly-games.appspot.com/maze?lang=it)

Some videos were initially screened on the interactive whiteboard and some examples puzzles were discussed and solved collectively. Later the children were invited to continue the path individually at home during the weekends or at school during the hours of the labs with the mates' help and the teacher's advice.

3.2 Using Scratch for the free development of personal projects

Usually the presentation of the Scratch environment has been very short and simple.

At the beginning of some sessions using the interactive whiteboard the teacher showed the students the development of a simple project that the children had to repeat step by step.

This "introduction" has never exceeded the length of 10 - 15 minutes.

Later the students were asked to develop their own original design or modify the example according to their taste or even to explore the Scratch website looking for projects to remix.

This 'free' activity has occupied most of the hours dedicated to labs. Cooperation among children, exchange and mutual support have been strongly encouraged. Even the use of a personal account in the context of the social environment of the site has been encouraged.

Lastly the printed and laminated Scratch Cards (https://scratch.mit.edu/help/cards/) were made available to students. Similarly some copies of the Getting Starter Guide (https://cdn.scratch.mit.edu/scratchr2/static/__90f8d4d8afbc51e3d823ca5efcc0ea53__/pdfs/help/Getting-Started-Guide-Scratch2.pdf)

3.3 Using Scratch to build a collective project

In our experience, Scratch has often been used as a tool for the implementation of projects related to curricular subjects or even when it was necessary to build something for a major event of the school life.

Children made animated backdrops for their theater performances, an interactive installation for Sciences exhibition (this one using a Makey Makey board - http://www.makeymakey.com/) and especially created animated stories with storytelling. Cappuccetto Rosso 2.0, which you can see in the appendix, is an excellent example of what we mean by "collaborative project"

Works on group projects has, however, involved the totality of pupils in classes.

The teacher, in this type of activity, has obviously taken on a more managerial role. He had to deal with the organization and coordination of teamwork, assigning modules or specific activities of the project (search for information, graphic design, implementation of sound or music, final assembly of the parts).

All above activities have been performed by schoolchildren 7-10 years old.

4 Coding strictness and written language teaching

Coding activity is very useful to introduce to the written language formalism, especially when pupils are foreign students and not native speakers. In a context of predisciplinary teaching, typical in primary school, our aim was to use the absolute stringency of the code (always verifiable through feedback of the computer) so that students would understand better the need of grammatical rules in written language.

During the school-year 2013/2014 we faced many pupils that aren't italian mother tongue even if they are italian native. In the primary school, teachers usually talk and read a lot to improve communication skills and vocabulary, but at the end the need of learning the written language rules arrives. Transition from oral language to written language is not so easy as it could seem at a first sight.

In the 5th grade of De Amicis school in Turin, we planned to use coding to make pupils easier to understand the need of rules for the written language. For the activities here described Scratch was the software for our pupils!

A similar intuition was already present in an educational robotics project described in [c] where pupils were using an Italian Logo-like language to program their Lego robots. In both cases teacher's target were:

- introduce pupils to a formal environment, i.e. an environment with strict rules pleasant and near to their way of thinking
- using of computers to improve motivation and to avoid the direct teacher-pupil interaction
- start using a new language, formally strict, but with immediate feedback
- developing comparative discussion about Scratch written language and italian written language.

Activities lasted an entire school year even if they were not so regular: sometime we worked more on Scratch; other times we worked more on written text, studying the syntax of the italian language.

Scratch tasks were not only focused on coding and developing problem solving skills, but the target was to learn about how strict can be a written language, whatever it is.

According to a choice of a non-directive teaching practice, children were invited to translate the code's commands in natural-language sentences (Italian).

The absence of any (traditional) interaction pupils-teacher but only pupils-pupils and pupils-computer strengthened this target.

The path was positive and the results were very good: pupils were motivated and the return on written language was encouraging. Every pupils understood that each written language needs fixed rules.

Differently from previous activities identified in paragraphs 2 and 3, this activity could not be included in a curriculum of CS, but it provides a good example of inter-disciplinary operation.

5 Conclusions (very temporary)

The activities summarized here led our group to reflect on the differences in educational approach to the issues of coding, programming and Computational Thinking, in an attempt to start to define a possible curriculum for primary school.

It is immediately evident the large difference between an activity with the 'puzzles' and a 'free' activity using Scratch. In the first case, the learning goals are gradually fixed from the outside (more from the website, not so much by the teacher). In the second case it is the student who sets himself its objectives, if appropriately encouraged. This situation puts the child in more cognitive challenge: 'imagine' and plan a possible solution to his problem.

On the other hand the activity for preset sequences of concepts (sequence, iteration, conditions, functions, ...) offers the teacher the guarantee to offer, in a gradual way to difficulties, all the key concepts that a good curriculum for the primary school should provide.

All this means, therefore, that, at least from our point of view, it is not possible, anyway, decide once and for all, what is the best path. There are conditions, in everyday school life, quite variable from one situation to another. What may be advantageous in one, may not be so on another.

Certainly the experience of both types of activity at different times seems appropriate and advisable. For instance, we noticed that the children who had already experienced the path of Intro course Code.org, showed greater confidence and mastery in dealing with the planning of their Scratch projects.

Even unplugged activities can be integrated in a complementary way with other kinds of experience. The unplugged activities, especially in the initial stages of learning a concept - key, especially with younger children, can be an indispensable cognitive aid when, in live situations, you replicate the issues raised by digital puzzles.

[a] H. Abelson, A. Di Sessa, Turtle geometry, 1986

[b] T. Filippini and V.Vecchi (eds.) The one hundred languages of the children , Reggio Children Publisher, 1996/2005, 216 pagg., ISBN 978-88-87960-08-2, Reggio Emilia.

[c] G. Barbara Demo, T4T: a peer training model for in-service teachers, WiPSCE 2015, Berlino

[d] G. Barbara Demo, G. Marcianò, Contributing to the Development of Linguistic and Logical Abilities through Robotics, Conference EuroLogo 2007, August 2007, Comenius University Editor, Bratislava

[e] S. Papert, Mindstorms: Children, Computers, and Powerful Ideas, 1980

[f] A. Rabbone, Bambini che imparano a programmare, Blog (http://bambinicheimparanoaprogrammare.blogspot.it/)

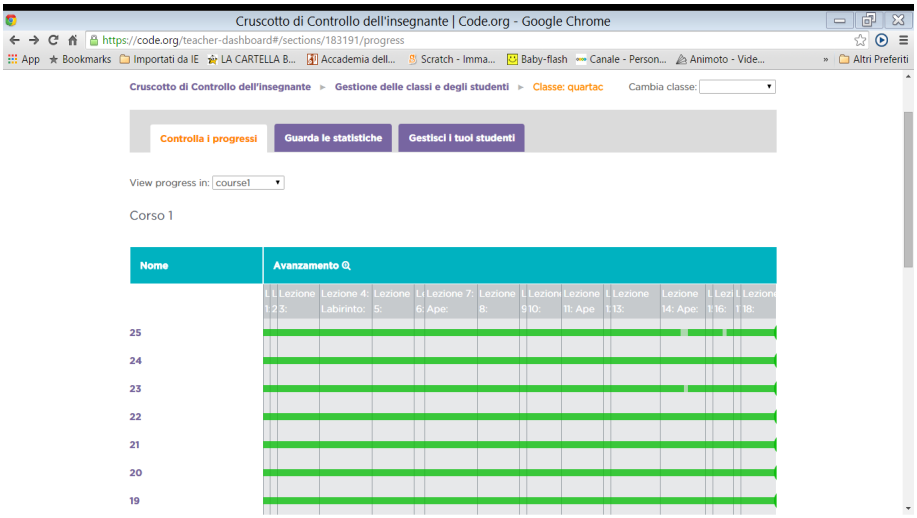
[g] B. M.Varisco, Nuove tecnologie per l'apprendimento, Garamond, Roma, 1998

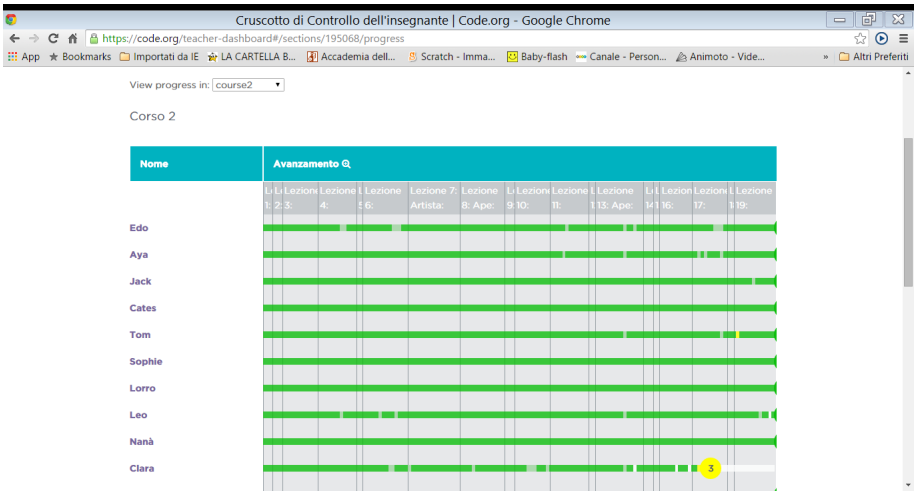
APPENDIX

RED RIDING HOOD 2.0

The 4th grade class of I.C. Tommaseo primary school began work on this project to participate in the Samsung Smart Code challenge 2015. This challenge focused on promoting coding and computational thinking through digital or analogical (unplugged) work.

During the school year the class experimented with the Scratch platform and many of the pupils enrolled with the community as individuals. Their work did not go further than attempts at sprite management or cartoon style scenes Apart from the lack of individual know how, after the general introduction to the class, the tool was used in spontaneous play. In December 2014 the teacher enrolled the class in the "Programma il futuro" initiative marking the start of a new way of working. All the pupils were enrolled on some "code.org" course and were invited to complete the work both at school and at home with the provision of free and collaborative periods. In a truly constructive way, the more competent pupils helped those with more difficulties to find solutions and finish the work whilst the role of the teacher was that of stimulus provider and moderator. Familiarity with block programming, the opportunity of guided work and the fact that there were objectives to be achieved meant courses 1 and 2 were soon completed and fairly accurately.





We then progressed to the 20 hour course which is under way at present. The whole class divided into two parallel groups is now working as a tutor for other classes that want to follow the "code" experience sharing both the equipment and the skills acquired.

The pupils have thus learnt how to block programme as well as ensure safety and automatism on simple programming operations, for example the awareness of different points of view.

Taking part in the Samsung competition has provided the opportunity to test technological and non technological class skills in the construction programming and creation of an imaginary product. At the early stages the class went through a brainstorming phase on the "Padlet" (virtual wall); they abandoned the idea of a videogame in favour of videonarrating a fairytale. The class had previously been involved in a cartoon creation workshop (Calimero which is currently showing on RAI TV) and thus knowing about Propp functions, storyboard and storytelling they favoured animation. In view of limited time they chose to interpret the well known fairytale of Little Red Riding Hood rather than their own invention. Using a calendar we calculated that we had just four weeks to dedicate exclusively to the project (interrupting the teaching programme) which worked out to be about 60 hours school time. At this stage we identified the necessary roles to carry out the animation and then planned each step. Taking into account the children's aptitudes and preferences we identified the following roles: Documentarists (those recording each phase and updating the scrum board), Screenwriters (those drawing up the paper storyboard and the dialogues), Scenographers (those drawing up or reelaborating the backgrounds), Phonics (those choosing and working on the soundtrack and the voices), Costume designers (those working on the costumes), Photographers (those documenting and photographing the sprites), Programmers (those mounting the scenes and the movement), the Special Effects Group (those elaborating the photographic and animation effects).

The groups worked autonomously and all the teacher had to do was coordinate the various phases. Decisions were taken collectively and the mounting done in real time

on the whiteboard meaning the whole class was involved and often useful comments were made to find the best solutions. The project was beneficial in learning how to cooperate. It provided motivation to reach a common objective within a deadline. There was a true atmosphere of cooperative learning, relationships were improved and the pupils became individually and collectively responsible, peer education was enhanced as well as emotional skills. The class explored traditional and technological tools in order to adopt the best strategies to reach their objectives of quality, usefulness and aesthetics without neglecting the audience's emotional involvement in the final product. The use of internet, different types of software, personal devices together with the school premises and tools provided a modern and effective learning environment as well as food for thought over new initiatives. Experience in programming made it clear that the application of computational skills and thinking within school is decidedly useful both in daily problem solving and for the planning of projects and new initiatives.

Photos of the activities