Metadata of the chapter that will be visualized in SpringerLink

Book Title	Informatics in Schools	
Series Title		
Chapter Title	And Now What Do We Do with Our Schoolchildren?	
Copyright Year	2016	
Copyright HolderName	Springer International Publishing AG	
Corresponding Author	Family Name	**Demo**
	Particle	
	Given Name	**G. Barbara**
	Prefix	
	Suffix	
	Division	
	Organization	Dipartimento di Informatica – Università di Torino
	Address	Corso Svizzera 185, 10149, Torino, Italy
	Email	barbara@di.unito.it

Abstract

After training activities introducing computing, many teachers seem to ask themselves “And now: what do we do with our schoolchildren?”. When we had to design a course for about twenty-five teachers, from twelve primary and middle schools, our first concern was trying to change this. For about twelve hours of the course meetings we introduced programming, then we discussed with attendees three frameworks of long programming activities: telling stories, creating group and class quizzes and relative answers, inventing riddles modeled by equations. The participants were asked to invent, design and implement an activity adapting to their students one of the frameworks discussed. Some were able to actually work with their pupils in schools, at least partly, during the course as we invited. This approach is of interest in other teachers training courses since the suggested activity frameworks can be inflected with contents that schoolchildren are dealing with in school. Also, the teachers are helped overcoming their apprehension in proposing a first long activity to their students because they can carry on such an experience during the training.

Keywords (separated by '-')

Teachers training - Primary and middle school - New curricula - Questioning - Activity framework

And Now What Do We Do with Our Schoolchildren?

G. Barbara Demo(✉)

Dipartimento di Informatica – Università di Torino,
Corso Svizzera 185, 10149 Torino, Italy
barbara@di.unito.it

Abstract. After training activities introducing computing, many teachers seem to ask themselves "And now: what do we do with our schoolchildren?". When we had to design a course for about twenty-five teachers, from twelve primary and middle schools, our first concern was trying to change this. For about twelve hours of the course meetings we introduced programming, then we discussed with attendees three frameworks of long programming activities: telling stories, creating group and class quizzes and relative answers, inventing riddles modeled by equations. The participants were asked to invent, design and implement an activity adapting to their students one of the frameworks discussed. Some were able to actually work with their pupils in schools, at least partly, during the course as we invited. This approach is of interest in other teachers training courses since the suggested activity frameworks can be inflected with contents that schoolchildren are dealing with in school. Also, the teachers are helped overcoming their apprehension in proposing a first long activity to their students because they can carry on such an experience during the training.

Keywords: Teachers training · Primary and middle school · New curricula · Questioning · Activity framework

1 Introduction

The Scuola2.0 project was promoted in 2014 by the Municipality of Torino (Italy) for improving digital competencies in primary and middle schools. One of the components of the initiative was the request to organize a pilot course during the school year 2015–16 for about twenty-five in-service teachers. From our previous retraining experiences we knew to have several constraints: most of the in-service teachers have little time for retraining and, even younger ones, know little about computing. Almost all only have experienced digital literacy activities, i.e. they have been using the web and computer tools specific for an educational purpose [5]. Other peculiar constraints of the project were that only few of the teachers were volunteers and that we had to cut into two phases the meetings having half of them during autumn 2015 and the other half from beginning March 2016 to the end of April. We ended up organizing ten meetings of three hours each.

A. Brodnik and F. Tort (Eds.): ISSEP 2016, LNCS 9973, pp. 1–12, 2016.
DOI: 10.1007/978-3-319-46747-4_10

In our country there is no mandatory curriculum for informatics in education neither commonly accepted suggestion when optional activities are possible. There are several formal and informal proposals going from educational robotics to programming (using different languages), from *CS Unplugged* to *coderdojo* and *fablab* type of activities, naming just some of the most popular ones. Also when deciding the contents of retraining projects we need to consider recent criticisms toward the digital presence in education till nowadays coming from different sources, for example OECD [8].

After having for years experienced the above mentioned approaches, we support introducing basic principles of computing by programming as a contribution to defining new digital curricula for schools. With our proposals we are then in line with the suggestions by Schulte in [11], Ben Ari in [1] and with those coming from the English National curriculum for primary school where we read:

> The role of programming in computer science is similar to that of practical work in the other sciences: it provides motivation, and a context within which ideas are brought to life [2].

Moreover, our project concerns k-8 education: thus a practical programming work complies with Piaget's theory where we have that the "concrete operational" stage of children cognitive development is peculiar of the primary school age. Practical experiences also facilitate in-service teachers who must be introduced to computing with short courses. As for the type of the activities to be proposed we have in mind also Martha Nussbaum's concern on the contraction of the humanistic component in the curricula contents:

> More and more often we treat education as if its primary goal should be to teach students to be economically productive rather than to think critically and to become informed and empathetic citizens [7].

The Italian National Indications for k-8 education appear to share Nussbaum's concern.

The introductory programming environment we propose for all ages is Scratch because of the reasons summarized by Shneiderman [12] and by Romeike [10] to characterize programming development environments suitable for introductory experiences.

With all the above motivations in mind, our proposal for the Scuola2.0 project was to devote the first five meetings to introduce unplugged activities plus elements of programming using Scratch and the remaining five to consider types of activities that the attendees could immediately implement in school adapted to their pupils while our lectures were continuing. Previous experiences had shown that at the end of retraining courses many of the participants had not felt able to design articulated activities adapted to their classes or, not having colleagues in the same school to share the experience with, they had been afraid to tackle alone a first field-activity. For this reason often we saw replicated some of the examples and exercises from a programming course with results not always connected to each other and with the rest of the teaching. We asked the

attendees to implement one activity suitable for their students choosing out of three types of activities that we presented and discussed with them. The activity had to be developed while the second part of the meetings was going on so that the problems encountered could be discussed in person with the other attendees and the lecturer. After the end of the course, the virtual community environment of the project is still available for discussions but a starting phase where problems are discussed in persons is necessary, again from our past experiences. The three types of activities proposed are:

- inventing a narration that would gather the most relevant aspects of a topic covered in school or that might be of interest to the schoolchildren,
- inventing quizzes choosing a curriculum topic and deciding, first in group then with the entire class, a set of questions with multiple-choice answers better representing the topic,
- for middle schools: inventing riddles each modeled and solved by a linear equation.

We offered programming integrated with unplugged activities of the type suggested in [4] because we have closely worked for years with teachers in schools becoming quite respectful of their competencies in the pedagogical and methodological components of educational activities.

Here we describe how the lectures of the project went on. In Sect. 2 the general motivations are briefly resumed with a summary of the first five meetings. Sections 3 and 4 concern last five meetings. First story telling and riddles invention activities are described. Section 4 is entirely dedicated to the activities of questions & answers, organized as group and class quizzes, whose proposal and materials have been methodically organized more recently than the others.

2 Key Principles of the Project and First Meetings

In primary education activities count for what children acquire during the process of developing an activity as much as for the result produced. Digital experiences also should take into account this kind of methodological approach.

2.1 Scuola2.0 Principles

The key principles inspiring the activities suggested during Scuola2.0 are the following:

- every activity, including programming, shall be a learning environment contributing to the overall growth of the child in its ethical, social and intellectual capabilities, from the beginning to the end of the activity development,
- every action must have a specific educational goal and be integrated to the overall pedagogical and disciplinary contents of the grade it is proposed to,
- particularly in the early years, programming must be conceived as one of the "hundred" languages children shall use to create and express themselves, as from Loris Malaguzzi of Reggio Emilia schools [3].

The project could count on ten meetings, three hours each. During the first five meetings we introduced attendees to the *CS Unplugged* activities presented in [2] and to basic programming concepts, using Scratch, shortly summarized in this section.

2.2 Unplugged Programming

We have been introducing programming for years with activities that often were sort of an easier version of those present in first programming courses at the university or in technical upper secondary schools. Soon we felt mandatory to offer different activities more integrated to the educational contents and pedagogical methodologies particularly, though not only, in k-8 education. Thus we began the Teachers for teachers (T4T) experience where we work with teachers and collect suggestions from the field. We revisited Logo activities and *CS Unplugged* activities developed in schools. In [4], three primary school teachers of the T4T group describe various types of computer-related activities they have created with their pupils. For first grades of primary schools, they have experimented *CS Unplugged* activities, for example those moving a human-robot. The latter are activities on a school chessboard-like playground or similar where a pupil moves from one square to another one according to the instructions her/his mates give. Only four instructions are available at the beginning (`forward`, `backward`, `turn-left` and `turn-right`), then the instruction set is gradually enriched, for example with instructions for bringing something from a square to another one. Also pupils are requested to perform different activities such as:

– comparing the different paths obtained from different sequences of commands,
– comparing lengths of instruction sequences written by different groups.

The presence of an obstacle on the playground, in one of our schools there is a slide, enriches the possible activities since children must avoid the obstacle. Also: first writing down inside the school the instructions for a path, then verifying them on the playground, makes teachers and pupils concretely see the sequence of commands and better catch the concept. Besides, having only few lines where writing the sequence often generates the idea of parameters, `forward(n)` for example, or `repeat(n)`. This is the same ruse used in other environments, for example in Lightbot, https://lightbot.com/). Also, attendees shall find out that the human-robot written sequence of instructions corresponds to the sequence of actions we perform in some real world situations, for example similar to the sequence of actions written on the Fire Alarm Table, i.e. the actions we (must) perform when we hear the fire alarm in school. Learning achievements during unplugged programming make easier the activities that follow.

2.3 Plugged in Activities During the First Part of Scuola2.0

Like many authors recommend, programming can be present in k-8 education using an environment suitable to the age of the students. Besides, as we wrote

earlier, we shall propose suitable activities. Alessandro Rabbone with his pupils developed MicroWorlds activities such as those entitled "Let's sing" ("Si canta") and "The auger" ("La trivella") during the 2004–2006 project KidsIdeasActivities (BambiniIdeeProgetti) whose final video can be seen at http://win.rabbone.it/_irreMMjr/progetti.asp#. The mentioned titles are self explaining and suggest that the relative activities are quite different from those one can find in a university or technical school course for programming. We also saw stories in Alice that some teachers had developed in secondary schools. The Scratch workshop led at ISSEP 2011 by Katarína Mikolajová and Martina Kabátová [6], and Lawrence Williams' visit to our department in 2013, who showed us several stories in Scratch [13], brought materials to our idea of changing the kind of activities we were proposing in our projects going on using Scratch.

An introduction to basic programming concepts by writing easy stories using the Scratch environment was given during our first five meetings of the Scuola2.0 project. A story telling activity allows pupils to express their creativity whether using digital tools or not. Using development environments such as Scratch this activity can be done at very different levels of familiarity with the tool, see again [13] and its references. For this reason Scratch is often proposed in courses introducing computing.

First we show on the big screen a story whose code, not considered at the beginning, is a sequence of actions only. Then we look inside the code and disassemble it asking attendees to find components of the story we just saw that is, using the theatre metaphor, they find actors/sprites, backgrounds, the costumes changed by the actors, the songs that are produced. Attendees start doing something of their own by changing costumes, dialogues. Then they continue with reviewing the synchronization among actions and so on according to the principle of remixing recommended by Resnick and the group of researchers authors of Scratch [9]. While developing stories, basic programming principles are recalled from previous *CS Unplugged* activities or are newly introduced together with some achievements from the actual use of the tool:

1. command sequences,
2. very simple repeat (repeat n times), typically to move a sprite or changing backdrops one after the other,
3. synchronization using seconds (because it is the easier to begin with) by designing a timeline of the story,
4. some interactions, for example to ask the user's name in order to personalize the execution of an activity.

The "story telling" pattern is suitable also for schoolchildren who can barely read and write and is interesting because it can smoothly evolve toward stories requiring a long time for the design and for planning the several activities to produce the narration such as the drawing of the sprites and of the backdrops, deciding the dialogues, and so on.

During the interval between the first and the second part of the Scuola2.0 meetings, some teachers were able to develop *CS Unplugged* activities with their

pupils. They recognized patterns of commands used in those activities within the Scratch scripts and were more confident than the other teachers in reading the scripts of the first Scratch stories.

3 Long Activities to Be Proposed in Schools

Revising a curriculum trying to maintain a satisfying coexistence among old and new contents is a difficult task. Those who propose to introduce digital competencies in education assume a great responsibility on the one hand with respect to the contents of the other disciplines that are declined to make way for new contents, on the other hand with respect to time and money resources that are diverted to the new activities. In choosing the types of activities to be proposed during our project one of our intents was to conciliate the contents already present in k-8 education with the digital activities proposed to teachers and students. The first two activity-frameworks we proposed allow to begin with very simple activities yet introducing some programming principles and then continue with gradually increased complexity.

3.1 The "Story Telling" Pattern

As we already said, the "story telling" pattern is interesting because it can smoothly evolve from short plain narrations, sequential in their digital implementation, toward long stories involving an entire class as the "Red Riding Hood" tale produced in Scratch by fourth grade children [4]. This activity had many educational components equally important as the acquisition of computer skills. Think of the design components, the planning, the collaborative work, the definition and organization of contributions, the timing and verification of the results, the children's feeling of responsibility for finishing in time the work assigned to them: all this next to the digital implementation of the story [4]. But here we shall focus on the five meetings whose goal was to make attendees design and implement a Scratch activity with their schoolchildren while attending the project. Thus what happened in schools and possible problems could be discussed and solved with the other teachers and the lecturer.

Obviously, the discussions during the second phase of the project also gave way to enrich the knowledge and experience on programming acquired during the first meetings, in particular to solve the problems arising from the inventiveness of the schoolchildren (beginning with the typical in the field programming problems: for example those concerning how to delete a figure background to create a new sprite). The Scuola2.0 meetings had the role of organizing the different activities, defining new steps, with colleagues' suggestions, and receiving help with respect to the problems found while developing such activities in the school.

Some forms of interaction in story telling often lead to the idea of developing quiz activities. In this way we have a smooth transition from storytelling and easy types of quiz activities. Which also means a smooth introduction of variables for storing scores or remembering errors.

One of the narration activities was developed by a teacher with her second grade pupils. It concerns animals and environments where they live. Children decided to have: the house, the forest, the sea and the savanna and drew appropriate backdrops. They also drew some animals for each environment, for example a red fish for the house, a lion for the savanna. First idea was to have animals appearing on the screen, each with the proper backdrop, saying something about its life. But the teacher during our meetings liked better to have an activity to practice English. Thus she and a second teacher implemented a quiz where, when an animal appears on the screen, the child sitting in front of the screen enters the animal's English name. For this first attempt children drew the backdrops and the animals. Thus the result is only partially developed with children. But for the Scuola2.0 project we consider positive that a teacher who knew nothing of programming at the beginning of the project, after six/seven meetings introducing her to computing had the initiative of proposing to her pupils this activity. She had to organize her schoolchildren deciding with them the four environments, and then divide pupils in groups, each group working on four animals one for each setting. This teacher's idea is to ask pupils to modify current year activity letting pupils dig into the code and producing something of their own at the beginning of the next year when they will be in their third grade.

Another Scuola2.0 attendee worked with his pupils on a long story about myths of ancient Greece with different components each developed by a different group of pupils. During our meetings we discussed about an easy way of putting together the components and together we found how. Among problems to be solved we had the question of global variables, i.e. variables every sprite can see, whose copies were all maintained when the projects developed by each group were integrated in a single activity. All these conquests came from working contemporarily in schools and with colleagues during the Scuola2.0 meetings.

Going toward a quiz activity, even if very simple like the one on animal names, programming concepts introduced are:

1. *selection*: for verifying the answers,
2. *variables*: introduced if we want to count the score,
3. *repeat until condition*: possibly introduced depending on the type of quiz (in the case of the animal names, *repeat until* "the answer specifying the name of the animal is correct").

3.2 Inventing Riddles

Retraining courses using Scratch are very well received by teachers because they appreciate the use of a simple tool through which students can get rewarding results. Also teachers appreciate they are asked to work on activities fit to their students and with contents that can be interesting for other disciplines and then for fellow teachers. Examples of interdisciplinary activities are the programs "think a number (and I guess it)" in which each group of students invents its own riddle through an experimental activity on linear equations. Here we refer to a type of riddle popular in our country played between two students s1 and s2. An example of this kind of riddle is the following list of requests:

1. s1 to s2: - think a number (let's call it x)
2. s1 to s2: - add 7
3. s1 to s2: - multiply by 3
4. s1 to s2: - subtract twice the number you thought
5. s1 to s2: - add 4
6. s1 to s2: - finally divide by 5
7. s1 to s2: - what number you end up with?
8. *s2 to s1*: - I have 9
9. s1 to s2: - thus, you thought 20!

This riddle corresponds to the equation:

$$(3(x + 7) - 2x + 4)/5 = 9$$

then $3x + 21 - 2x + 4 = 45$ and $x = 20$.

We can write a Scratch activity where the student s1 is a Scratch sprite.

Let us call a the answer in 8 of student s2 to the question in 7 of student s1. In such activity $x = 5a - 25$. With answer 9 at point 8 we have $x = 5 \cdot 9 - 25$.

But you can also have another type of requests where x is cancelled. All requests are from student s1 to student s2 as in the following example:

1. think a number (again we call it x)
2. add 7
3. multiply by 2
4. subtract 4
5. subtract twice the number you thought
6. finally divide by 5
7. You got 2! How come I can guess right?

This riddle corresponds to the expression, where variable x disappears:

$$(2(x + 7) - 4 - 2x)/5$$

and 10/5 is the final value.

In both cases the corresponding Scratch activities are very simple: each is a sequence of instructions for conducting the dialogue. The valuable part of the activity is once again, and particularly in this case, the phase where the students begin playing the riddle unplugged, are asked to build a riddle of their own which can be done when they understand that a riddle can be modeled by an equation. And then they invent other riddles.

4 Questioning

The questioning activity has been suggested by various sources in the pedagogical literature where questioning techniques are largely discussed [10]. Not least the fact that a Scratch activity designed as an exercise to introduce the variables (in this case the "score" variable) in a quiz has been very well received in all

courses in which it was proposed because considered appealing for schoolchildren. Discussions with teachers, involved in the Scuola2.0 project and outside, has also shown that a type of very simple exercise such as a quiz has computing value in introducing a gradual use of the variables but also offers original ways of learning: each group of students creates its own quiz where questions are proposed on the topics of a lesson that most affected the group. These are examples of the many activities that promote the involvement of teachers in other disciplines and thus the gradual upgrade of the digital skills of these teachers also: involvement essential if you want to get to use the digital as a tool for constructively learning various disciplines.

The questioning activities we propose will have two phases:

Group work: with groups of two or three pupils per group. During this step each group produces a quiz with 3 or 4 questions on a topic, that can be a curricular one. We chose to suggest having multiple answers to increase the work by each pupil in the group. During this phase the members of the group review the topic and decide which are its representative components candidate to become matter of queries. Similarly the relevant answers to the chosen questions are decided to build a multiple-choice quiz. The group-quiz is finished with images and sounds;

Class work: at school each group shows to the entire class questions and answers present in its group-quiz. The goal is to produce a class-quiz from group-quizzes with more questions. Questions and answers are chosen among those proposed in the group-quizzes, both can result from different formulation of what contained in group-quizzes to take into account aspects deemed important by the children or according to teachers' suggestions.

The class-quiz can have various uses: for example, can be shared with another class having the same subject in curriculum. This will assess the pupils' ability both to answer questions and to evaluate possibly missing important aspects of the considered topic.

Positive aspects of this activity are:

- the discussions that are developed on the topic,
- the active learning aspect,
- the possibility to involve all students. In the digital implementation maybe you can choose to put in the class-quiz at least the figures of a group or of a student disappointed with respect to questions/answers, for example because the ones he has proposed have been discarded.

Working to build a quiz stimulates a series of reflections and activities that involve different learning areas. As in the story telling, the planning phase requires the choice of a topic and the definition of the objectives, the selection of materials and the definition of tasks and deadlines, then the organization of a scrum board. Later and during the development of the activity, critical comparisons are necessary and then, a self evaluation test at the end of the activity on the strengths and critical points of the process and of the result. Figure 1 shows

a screenshot of a quiz on neurons (from an activity by Carlotta Craveri, student of Educational Sciences in our university during the academic year 2014–15). The figure shows the question "How do neurons communicate?" whose proposed answers are:

A. Using smoke signals
B. Using WhatsApp
C. By means of the synapses

This example has been shown to the attendees of the Scuola2.0 meetings. In fact for each activity type, because of the short time available, it was decided to show and discuss with the teachers a framework, i.e. a very simple yet running example for each one of the proposed activity, and then let them start working on such frameworks.

Fig. 1. How do neurons communicate?

5 Conclusions

This is not a contribution toward defining digital competencies needed to future k-8 teachers. It is a contribution to deciding how in-service teachers can be trained to gradually gain the competencies for developing with their pupils the digital activities we need to offer in k-8 education without further delay. Of course it requires we define, at least partially, what we consider the proper digital curriculum for k-8 education so that we can go toward the chosen direction.

Here we shortly report what has been the computing training for a group of k-8 teachers. The paper particularly concerns the second half of the training meetings where we pushed the teachers to develop with their schoolchildren an activity lasting for the two last months of the school year (April and May in

our country because June is to be left for conclusive verification activities). The teachers were suggested three types of activities aiming to have all attendees working on a program that lasted for several weeks.

The seventeen primary school teachers attending the Scuola2.0 meetings have all chosen the story telling activity that is in fact an activity feasible at very different levels of difficulty: one can start by modifying a given story that, in our case, is the given reference framework. Instead, teachers in middle schools judged feasible in their classes all three kinds of activities but, perhaps because they are all mathematicians, they focused on stories and riddles (the latter also because more directly related to mathematics) motivating the choices with the short time available in this end of school year 15/16.

The operating methodology we chose provided the course attendees with a framework of each of the suggested types of activity, i.e. a working simplified version of it. According to the participants this method is proving to be very useful. Also useful the Moodle community where participants could ask for help with problems and discuss solutions during the two months when the meetings took place. The community is almost essential after the end of the course. Not all teachers could attend the last meetings and only few of the attendees could actually develop almost entirely the chosen activity with their pupils. From the meetings assessment, the teachers motivated this failure with their short and busy time when the end of the school year is approaching. As we wrote, digital activities in primary school are learning environments to be entirely developed from the design phase without aiming for the final result only. Thus the request of the teachers to have all the meetings in the very beginning of the school year could be a way to increase the actual transfer of their experience inside their schools.

We do not have an assessment of our experience yet, but we have seen teachers curious of implementing something that on the one hand they felt near and useful for what they were doing every day, on the other hand new and fitting the direction schools shall take in the future. Also, the teachers considered quite positively the narrative and the question/answer frameworks as a continuation of each other. This was perceived as having an activity to develop with pupils lasting for several sessions and consequently not an occasional exercise that is one of their frequent complaints. The second half meetings turned out to largely be a discussion time for solving implementation problems, for reciprocally showing the activities, exchanging ideas and pedagogical comments, acquiring new abilities from colleagues' suggestions. The general feeling was that the activities developed during the meetings helped overcoming the teachers' fear of inventing something of their own. The integration of educational content specific to the class where an activity is proposed is an outcome of the active involvement of the teachers.

An open question concerns the MOOC with the same contents of the course here described, that the Municipality recently requested us. Our doubts on its effectiveness come from considering that the in person meetings gave a fundamental contribution to the success of this course.

References

1. Ben-Ari, M.M.: In defense of programming. ACM Inroads **7**(1), 44–46 (2016). doi:10.1145/2827858
2. Berry, M.: Computing in the national curriculum. A guide for primary teachers. Computing at School (2013). http://www.computingatschool.org.uk/data/uploads/CASPrimaryComputing.pdf
3. Cagliari, P., Castagnetti, M., et al.: Loris Malaguzzi and the Schools of Reggio Emilia. Routledge, Abingdon (2016)
4. Ferrari, F., Rabbone, A., Ruggiero, S.: Experiences of the T4T group in primary schools. In: Jekovec, M. (ed.) The Proceedings of International Conference on Informatics in Schools: Situation, Evolution and Perspectives - ISSEP 2015. University of Ljubljana, Faculty of Computer and Information Science, October 2015
5. Gander, W., et al.: Informatics education: Europe cannot afford to miss the boat. Report of the Joint Informatics Europe & ACM Europe Working Group on Informatics Education, ACM Europe, April 2013. http://europe.acm.org/iereport/ACMandIEreport.pdf
6. Kabátová, M., Mikolajová, K.: Fostering creativity through programming - scratch workshop. In: Bezáková, D., Kalaš, I. (eds.) ISSEP 2011 Proceedings of Selected Papers. Comenius University, Faculty of Mathematics, Physics and Informatics, Bratislava, October 2011. http://www.issep.2011.org/files/workshops/w07.pdf
7. Nussbaum, M.C.: Not for Profit: Why Democracy Needs the Humanities. Princeton University Press, Princeton (2010)
8. OECD: Students, Computers and Learning. OECD Publishing (2015). http://dx.doi.org/10.1787/9789264239555-en
9. Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., Kafai, Y.: Scratch: programming for all. Commun. ACM **52**(11), 60–67 (2009). doi:10.1145/1592761.1592779
10. Romeike, R.: Three drivers for creativity in computer science education. In: Benzie, D., Iding, M. (eds.) Proceedings of the Working Joint IFIP Conference Informatics, Mathematics, and ICT: a 'golden triangle', IMCT 2007, Boston (2007)
11. Schulte, C.: Reflections on the role of programming in primary and secondary computing education. In: Proceedings of the 8th Workshop in Primary and Secondary Computing Education, WiPSE 2013, pp. 17–24. ACM, New York (2013). http://doi.acm.org/10.1145/2532748.2532754
12. Shneiderman, B.: Creativity support tools: accelerating discovery and innovation. Commun. ACM **50**, 20–32 (2007)
13. Williams, L., Černochová, M.: Literacy from scratch. In: Proceedings of the 10th IFIP World Conference on Computers in Education, WCCE 2013, pp. 17–27. Copernicus University, Torun (2013)

MARKED PROOF

Please correct and return this set

Please use the proof correction marks shown below for all alterations and corrections. If you wish to return your proof by fax you should ensure that all amendments are written clearly in dark ink and are made well within the page margins.

Instruction to printer	*Textual mark*	*Marginal mark*
Leave unchanged	··· under matter to remain	
Insert in text the matter indicated in the margin		New matter followed by or
Delete	through single character, rule or underline or through all characters to be deleted	or
Substitute character or substitute part of one or more word(s)	through letter or through characters	new character or new characters
Change to italics	under matter to be changed	
Change to capitals	under matter to be changed	
Change to small capitals	under matter to be changed	
Change to bold type	under matter to be changed	
Change to bold italic	under matter to be changed	
Change to lower case	Encircle matter to be changed	
Change italic to upright type	(As above)	
Change bold to non-bold type	(As above)	
Insert 'superior' character	through character or where required	or under character e.g. or
Insert 'inferior' character	(As above)	over character e.g.
Insert full stop	(As above)	
Insert comma	(As above)	
Insert single quotation marks	(As above)	or and/or or
Insert double quotation marks	(As above)	or and/or or
Insert hyphen	(As above)	
Start new paragraph		
No new paragraph		
Transpose		
Close up	linking characters	
Insert or substitute space between characters or words	through character or where required	
Reduce space between characters or words	between characters or words affected	