What's the Fun in Informatics? Working to Capture Children and Teachers into the Pleasure of Computing*

Violetta Lonati[1], Mattia Monga[2], Anna Morpurgo[1], and Mauro Torelli[1]

[1] Università degli Studi di Milano,
Dipartimento di Scienze dell'Informazione
[2] Università degli Studi di Milano, Dipartimento di Informatica e Comunicazione,
Via Comelico 39, 20135 Milano, Italy
{violetta.lonati,mattia.monga,anna.morpurgo,mauro.torelli}@unimi.it

Abstract. The importance of computer science education in secondary, and even primary school, has been pointed out by many authors. But too often pupils only experience ICT, both at home and at school, and confuse it with computer science. We organized a game-contest, the *Kangourou of Informatics*, with the aim to attract all pupils (not only the talented ones), expose them to the scientific aspects of informatics in a fun way, and convey a correct conception of the discipline. Peculiarities of the game are its focus on team work and on engaging pupils in discovering what lays behind what they experience every day.

Keywords: informatics and education, learning contests.

1 Introduction

"*I think that it's extraordinarily important that we in computer science keep fun in computing. When it started out, it was an awful lot of fun.*" The pioneer Alan Perlis (1922–1990) stated this very clearly as a priority goal, but today informatics is rarely depicted as a rewarding activity in itself. Although our lives are inevitably interwoven with computers, software, and automatic computations, most of the general public perceives informatics as technological overhead to be avoided as much as possible. In fact, informatics is great in producing ready-to-use abstractions that can be used as black boxes for specific tasks, but the abstract nature of computation is there, and it needs to be discovered to open up real innovation and human development. So we are increasingly facing the problem of convincing non-experts that the desirability of informatics is not just in its value as an instrument, but also as a human challenge, both intellectual and social. A basic understanding of the underlying principles of informatics should therefore be common knowledge among the general public as are the principles of other more traditional sciences. However, not everyone is convinced

* This work was partially funded by Google under the Computer Science for High Schools (CS4HS) program.

I. Kalaš and R.T. Mittermeir (Eds.): ISSEP 2011, LNCS 7013, pp. 213–224, 2011.

that teaching computing to kids is desirable or, perhaps, *feasible*. Indeed, informatics could play a very important role in education as it promotes the ability to deal with abstract concepts, the acquisition of the scientific method, and a pragmatic approach to problem solving and work organization [11,2]. Historically, as it became evident that the computer was beginning to play a very important role in our every day life (in 1982 "The Computer" was named Machine of the Year by Time Magazine), projects to introduce informatics in schools were developed. Informatics education started with programming, and the Logo programming language was used for a while for that aim. But as the personal computer made its way through in schools and homes, the attention shifted from informatics to the use of the PC and its applications: computers were sometimes bought by schools before developing informatics curricula and preparing teachers adequately. Most was left at the individual initiative and fantasy. The effect was a serious misconception of what informatics is about, and informatics is still nowadays too often confused for ICT [4]. Thus, even if the last decades have seen an increasing exposure of children to computer applications, the basics of information, computing, and software sciences are still not part of the curricula of primary and secondary schools in several parts of the world. According to a report of ACM and the Computer Science Teachers Associations (`http://www.acm.org/runningonempty`) the number of computer science courses in the US has decreased in the past five years, and when the schools offer a course, it is usually an elective one. Moreover, most of these courses are in fact about information technology literacy rather than a look-ahead toward the issues introduced by the automatic elaboration of information. In Italy the context is even worse: only computer literacy, mostly presented as an instrument to increase one's proficiency in other subjects, is indicated as an important educational goal, and even that is never taught as a specific subject by experts of the field (except for vocational, technical, schools). In most Eastern European countries informatics as a separate subject was established 30 years ago: training of programming skills used to get more attention at the beginning, while nowadays much more consideration is devoted to developing ICT skills [6].

1.1 The Many Facets of Informatics

Indeed basic computer application literacy is certainly a useful skill to acquire, but not at the cost of confusing pieces of knowledge that are largely independent: fluency in an application domain and familiarity with the computing disciplines.

Mirolo [13] points out that the term *informatics* is used in different contests with different meanings:

1. informatics as a science, providing its own peculiar key to interpret reality and its specific approach to problem solving;
2. informatics as a technology, concerning the characteristics, structure and working principles of the now ubiquitous hardware and software devices;
3. informatics as an instrument, providing practical tools to manage information in many different contexts.

It is not easy to give a precise and complete definition of what computer science is. Hromkovič [11] points out that computer science is for some aspects a meta science such as mathematics, as it investigates general categories; a natural science for other aspects, as it studies objects and processes, it deals with quantitative rules of natural processes, it investigates what is possible and what is impossible, and it uses the scientific method; it is a problem oriented and practical engineering discipline dealing with technical and management issues. It is therefore a challenging task to work for a correct perception of informatics. To clear the confusion from the ground up, we coined a specific word for the use of computer applications that does not require computing skills: we called that *applimatics* in [12]. Applimatics is not only the use of office automation tools, but also the use of a dynamic geometry program to explore Euclidean axioms. Even the use of things like the Alice [5] programming environment to explore the properties of bodies immersed in a 3D space and graphic animation techniques or the meritorious Logo are not, per se, informatics, if the emphasis shifts from computational thinking to geometry[1]. We liked the challenge and started the *Kangourou of Informatics* game-contest (`http://kangourou.dsi.unimi.it`), which aims at giving a correct view of computer science to both pupils and teachers, by exposing them to the scientific aspects of informatics, not considered in syllabi of most schools. The intention is to convey the correct conception of informatics in a fun way and attract all pupils, not only the talented ones. We identified, however, three main obstacles that have to be overcome to effectively propose computing games to children and (non expert) teachers. On the one hand there is the abstract nature of computing (mostly shared with mathematics), that makes it difficult to show the concepts by referring to physical objects and the fact that often specialized language and terms are needed; the games should avoid too abstract tasks and the use of jargon to propose the challenges (we sometimes propose jargon as a challenge in itself!). On the other hand, however, it is very important to avoid to completely disappoint the expectations of pupils about informatics; the link with applimatics to which they are daily exposed should be made clear as much as possible, to grow a consistent and fertile global picture.

In the following sections we will describe the main ideas that drove us in setting our game-contest: in Section 2 we briefly survey informatics competitions; in Section 3 we describe our experience in organizing the *Kangourou of Informatics*; in Section 4 we report the feedback we got from the participants; and in Section 5 we draw some conclusions.

2 Promoting Informatics through Competitions

Consistently with the idea that computing is fun when you engage in it, games and contests are rather common among computer science experts. The most famous contest is probably the International Olympiad in Informatics (IOI) [17,16]. The first IOI was held in Bulgaria in 1989 under the sponsorship of UNESCO. IOI is open to high school pupils throughout the world. The competition tasks

[1] Indeed Logo was originally proposed to improve the teaching of mathematics [15].

proposed to participants are of algorithmic nature, and, while they aim at stimulating interest in informatics and information technology, the explicit goal of IOI is to give recognition to young pupils from around the world who are the most talented in computing. On similar lines the ACM organizes an International Collegiate Programming Contest (ICPC) in which teams are given 5 hours to solve between 8 and 12 programming problems: the winner is the team which correctly solves most problems. Moreover, contests are increasingly common in the information security arena. In the '90s the DEF CON conference popularized a computer security wargame called "Capture The Flag" (CTF): each team is given a (virtual) machine or a small virtual network to defend from the attempts of intrusion of the other players. Teams are scored for both successful defense and attacks. CTFs evolved into full educational exercises to give participants experience in securing complex systems. In 2004 an international, academic CTF was started by the University of California at Santa Barbara, and several other similar contests are organized around the world and often attract online contestants from several countries and different educational backgrounds. These competitions, however, assume that participants have already acquired some expert skills in the field: an accompanying training is sometimes planned to bring the contestants to the level required to compete. IOI, for example, promotes local training sessions (often several weeks long) to teach high school pupils the programming skills needed for the Olympics. Thus, such competitions mostly address an audience that is already inside the fun of informatics or had at least the opportunity to see it from a technical point of view. This chance occurs either because of a meeting with an especially motivated teacher or, more likely, to pupils attending a specialized educational program. Thus, most pupils are out of reach of these contests, at least in countries (like Italy) where computing is not explicitly part of generic curricula: they often end up in knowing only the applimatics face of informatics, but the challenges of information technology are rarely presented to them.

An important step in popularizing the fun of informatics to a public with little knowledge of the technicalities of the discipline is the rise of non-specialized game-contests. In 2009 we started the *Kangourou of Informatics* by piggy-backing the experience of our University in organizing the *Kangourou of Mathematics*. In Kangourous the games have the explicit intent to attract the maximum number of pupils without aiming at any national selection nor at a comparison between countries and we embraced exactly those goals to foster the knowledge of computing to an audience as vast as possible. In fact, we recently discovered that Kangourous were inspiring several other people long before us. In 2004 Valentina Dagiene started a Kangourou-inspired game-contest in Lithuania under the name of *Beaver*[7,9,8,1]. The game became international in 2007 and it has several points of contact with our proposal: indeed we would really love to join our efforts. In the following sections we try to describe the peculiarities of our approach in organizing our game-contest and what we want to convey to pupils about our discipline.

3 The Kangourou of Informatics

A game-contest, the *Kangourou des Mathématiques*, was created in 1991 in France by André Deledicq on the model of the Australian Mathematics Competition, with the goal of contributing to the popularization and the promotion of mathematics among young people. The success was immediate also thanks to the associated distribution of a massive and pleasant documentation on mathematics to the participating pupils and their teachers. The French experience was exported abroad, first to Europe and then to other continents through an international association, *Kangourou sans frontières*, founded in France in 1995. The association's aim is to promote the spreading of a basic mathematical culture by all means and, in particular, by organizing the annual game-contest to be held on the same day in all participating countries. The game now counts millions of participants among primary and secondary school kids (48,000 in Italy in 2010). In Italy, which joined the association in 1999, the game is organized in cooperation with the Mathematics Department of the Università degli Studi di Milano. As a consequence of the effectiveness of the event, in 2005, at the yearly Kangourou Sans Frontières international meeting, the Romanians suggested to extend the game-contest to foreign languages and informatics. In 2008, Kangourou Italy invited two Informatics Departments of Università degli Studi di Milano, AICA, and SDA-Bocconi to study a formula for an informatics game-contest. On the basis of our previous experiences with IOI (as trainers for the Italian national IOI team) and UCSB International Capture the Flag (our team won in 2007 and and was placed among the first positions in 2008 and 2009), we took up the challenge. The first contest was held in 2009, and the third edition this year.

3.1 Organization of the Game-Contest

The *Kangourou of Informatics* is a team game-contest and is held yearly nationwide. Each team is composed of four pupils; at the moment there are two categories: category "medie" for junior high school pupils (age 11–13) and category "biennio" for pupils in the first two years of high school (age 14–15), with slightly diversified difficulty levels. We plan to extend the game also to younger pupils. There are two phases:

1. a qualifying round carried out on-line and organized locally in March in the schools under the supervision of the schools' teachers;
2. a final round in May for the best 24 teams, held in Mirabilandia, an amusement park near Ravenna.

In both rounds the problems are presented in a playful way, since one of the main goals is to make pupils enjoy themselves while discovering what informatics is. The game is mostly skill-oriented, as no prior knowledge can be assumed. Moreover, pupils are allowed and encouraged to use the Web in both rounds to search for information or hints they may need to answer the questions. The problems are at different levels of difficulty and spanning various aspects of informatics, from logic to programming, from grammars to concurrency.

The questions are chosen so that it is very unlikely to score zero points, but also very difficult to totalize the maximum score. The game-contest is partly self-supported with the subscription fees and has commercial sponsors which also provide some prizes for the first three teams in each category and their schools. After the qualifying round a booklet on the contest is published and sent to the schools. It contains the problems as presented during the game and, for each problem, the answer and how it can be obtained and the informatics topic to which the problem refers. Both the booklet and the on-line test are available on the Kangourou's website. The booklet should, respectively could, help teachers discover this discipline and overcome their fear of inadequacy regarding computer science (we refer to mathematics and technical education teachers, since in many Italian schools computer science is not taught as a separate subject). Besides it is intended as an aid in preparing for the final round and for the next year's competition both for pupils and for teachers.

3.2 Qualifying Round

The qualifying round is carried out on-line locally by the schools under the supervision of the teachers. The software and the problems are downloaded from a central server. The teams have to solve around 10 problems (see below for examples) in about an hour (it may vary slightly from year to year). The software keeps the time, collects the answers and sends them to the server for evaluation. We think the use of the computer in the contest, though not essential for the questions per se, is very important in order to keep the link explicit with what pupils and teachers normally associate with the word informatics. We believe it is very important to clinch to the fact we are not discussing something different from what the world expects from computer science experts: we are just trying to explain better what is under the hoods.

3.3 Examples of Qualifying Challenges

This subsection describes some of the games we proposed.

Maze. One of the problems was how to reach a treasure in a maze. The setting was a sequence of rooms, each one identified by a number, n, and with two exit doors, the left one leading to room number $2n$ and the right one leading to room number $2n + 1$. The goal was to reach room number 69, where a treasure was kept. The task was to guess the correct sequence of doors that had to be passed through to reach the target room. If an erroneous door was opened, the player lost one of his or her three lives and had to start over. The task could be solved with different approaches: by trial and error, by drawing the graph of the rooms' connections and finding there the path connecting the two rooms, by noticing that each room in this maze can be reached only from one other room and working backwards from room 69 to room 1: all even numbered rooms are reached by a left door, all odd numbered rooms by a right door. 20% of the teams (16% in the "medie") got the maximum score (no lives were consumed in

unsuccessful tries) and another 31% (same percentage in the "medie") were able to solve it with two or three trials.

A pipe and filter game. The problem stated that Riccardo has two lists of soccer teams: OldTeams and NewTeams. In the former he collects the teams of the 2010 edition of Champions League and in the latter the ones participating in the 2011 edition. He wants to know which teams participated for the first time, *i.e.,* which teams are in NewTeams but not in OldTeams. In order to compute the solution, Riccardo may combine three programs:

1. catenate, which is able to append a list to a given one;
2. dups, which returns a list of duplicates in a list;
3. uniqs, which returns a list of unique values in a list.

The game was proposed in two different versions for "medie" and "biennio". The easier one ("medie") asked to identify a correct solution among different sequences of program applications, expressed in natural language: the correct solution was "Catenate NewTeams to OldTeams; find duplicates in the result; catenate them to NewTeams; find unique elements in the result". The quiz was answered correctly by 40% of the teams. The "biennio" version asked to build a solution by combining the programs: each program was represented by a graphic block with changeable inputs and an arrow denoting the output as input *filtered* by the block. The goal was to produce a sequence by connecting a block to the preceding one: in fact all the blocks but the first had one input fixed to the output of the previous as in a pipeline of filters; the number of blocks was limited to a maximum of five. Although the pupils were driven by the graphical scaffolding, this version turned out to be much more difficult. Very few teams were able to find an optimal solution with only three blocks, and some others found a solution with 4 or 5 blocks (in total, a correct solution was built by the 19% of the teams only).

Secret sentences. In this problem a group of friends decided to defend their hiding place using secret sentences. To enter, one has to pronounce a valid sentence, which is built according to the following rules:

- valid sentence: a simple sentence or a simple sentence followed by an adverb, a verb, and a valid sentence;
- simple sentence: an article followed by a description of a mouse, or an article followed by a description of a cat;
- description of a mouse: "mouse", or a description of a mouse followed by "white";
- description of a cat: "cat", or a description of a cat followed by "fat", or "red" followed by a description of a cat.

The pupils had to identify which sentences from a given set were valid and allowed entrance into the hidden place. The topic dealt with by the question is the use of a formal grammar to generate valid sentences. The grammar is based on the pupils' natural language, but allows some counter-intuitive constructions: an adjective may follow a noun, and the same adjective may be repeated consecutively, thus the pupils had to be careful not to confuse the concept of correctness

under the rules of the game and correctness in their mother language. The question (proposed only for the "biennio" category) turned out to be quite difficult: only 15% of the teams solved the game perfectly and gained the full score. However, in total 92% of the teams were able to get some correct answers.

3.4 Final Round

The final round of the *Kangourou of Informatics* game-contest takes place in Mirabilandia amusement park and is reserved to 24 of the best qualified teams (at most one team from each school is admitted, in order to discourage cheating in the qualifying round). Participating in the final round is seen as a prize for all finalists: they are hosted with no extra expenses by Kangourou Italia and have a free admission ticket to all park attractions during the days of the finals.

The round is organized as follows. In the first evening a welcome meeting with all team members and accompanying teachers is scheduled, where they get information about regulations and the timetable. On the next morning all finalists are called for the actual contest that lasts the whole morning. During the contest, accompanying teachers are offered some lessons and labs, that suggest topics and innovative ways to convey informatics culture to pupils. The rest of the day is occupied with the grading process which is usually quite time-consuming. The day after the contest, a conference is organized for both finalists and accompanying teachers, which ends in the award ceremony.

The final competition consists of several games of different types. First, each team member is involved in a game aimed at obtaining its account to access a computer and an envelope containing the other assignments. These can be tackled in any order and have to be handed in all together at the end of the contest. Such assignments usually consist of written problems and questions, tasks to be developed on the computer, search tasks on the Web.

With the aim of stressing some peculiar aspects of informatics, we impose some rules that promote team work and smart use of limited resources: there is only one computer for every two teams and each session expires after 15 minutes. Moreover, computers are located at some distance from the place where team members are seated. Hence, it is important to manage some sort of team organization and time expenditure. The best teams often show an interesting subdivision of roles (the "programmer", the "logical thinker", the "pony express", the "checker", ...), some multi-threading of tasks (*i.e.,* combination of individual work, work in pairs, and discussions in the whole team), and a good ability in avoiding downtime (e.g., when the computer is unavoidable, time is not wasted by waiting, but it is used to establish which search should be done later on the Web, to design the solutions to be implemented, and to prepare instructions or entire programs to edit and execute later on the computer).

The initial game actually starts the evening before the contest, just after the welcome meeting: each team receives some material to read or to examine, which can be used to get some training for the competition. All team members get immediately excited by this unexpected preview and plunge into the material trying to catch it and figure out how it could be used! For instance, in the contest

of 2009, we divided each team into 2 subgroups, gave the team user-name to one subgroup and the team password to the other one, and asked them to exchange their information on a public channel using the cryptographic method described the day before, in order to complete the whole picture about their own accounts. In 2010, instead, the game was based on a riddle about trees, leaves, and heaps. This year, the game required to understand a non-losing strategy, described in a visual way, for tic-tac-toe. This part of the contest does not increase the score of the teams, but gives them the possibility to gain time over the other contestants. If a team does not succeed in solving the game within a fixed amount of time, the account information is revealed, a penalty is assigned, and the team is given the next assignments.

Written problems are quite standard, in that they focus on typical problem-solving issues in the field of data representation, cryptography, combinatorics, algorithms and data structures, languages (grammars and automata), games, logic, and so on. However it should be emphasized that problems are not in multiple-choice form. Usually open questions are asked and a brief explanation that motivates the answer is requested. The score assigned to this part of the test obviously takes into consideration the correctness of the answer, but also depends on other parameters: is the motivation relevant, accurate, complete, well-written?

The tasks requiring the use of computers usually concern some aspects of informatics we consider typical and educational.

Problem-solving and programming. We assume no prior knowledge about programming language syntax. Hence we propose visual programming environments than can be understood and mastered easily and quickly, like Logo [15], Scratch [14], Etoys [10]. The goals to achieve are also graphical: contestants had to build paths around obstacles in 2009, to draw colored geometrical pictures in 2010, to find a way of escape from mazes this year. A sequence of exercises is proposed, with increasing difficulty: first only sequential instructions are needed, then loops, sub-routines, and variables become necessary as the level increases. Some bonus points can be gained when a solution is correct with respect to several instances, in order to introduce the idea of generality of algorithms.

Text description and structuring. Teenagers are familiar with word processors that follow the *what-you-see-is-what-you-get* paradigm, like Microsoft Word or Microsoft PowerPoint. However, they do not have any idea about how these programs work internally, and have no model about how a text is represented by such kinds of software suites [3]. Moreover, they usually use these programs naively and often ignore the possibility to structure texts with *paragraph styles*, or to distinguish semantic structure from typographic aspects. This somehow corresponds to a deeper, logical, confusion between the meaning of the content and its appearance. During the final round, we always schedule a game that aims at discovering these aspects of text processing, asking to provide some written "formal description" of a structured text. In order to keep abstraction and concreteness in touch, we specifically designed and implemented web-based programs that were at disposal of the teams to test their solutions: such

programs use these formal descriptions (also if partial) to re-build the document. Thus pupils have the possibility to check what seems right and what is wrong, and they have the chance to correct mistakes, with a trial-and-error approach.

In 2009 we proposed the use of a *wiki*, *i.e.* the use of a simplified mark-up language that uses, for instance, quotation marks for emphasized text, marker "-" and proper indentation for itemized lists, and so on. The teams received a written tutorial about basic formatting rules with some examples and were asked to reproduce a full formatted text. In 2010 the game was titled "Into PowerPoint" and it asked for the description of a slide presentation in a tabular form. Contestants received a pdf with the slide presentation and a printed page with the set of all text elements occurring in the presentation. They first had to classify each slide according to a list of possible page formats. Then they had to fill in a table having a row for each text element, by defining the slide containing it and its position in the slide. Finally they had to describe some graphical aspects (text color, background color, thickness, alignment) of some basic elements of the presentation (titles, footer, ...)[2]. In 2011 we played with HTML: here the main goal was to recognize the tree structure of a given text, and mark each inner node according to its structural role (title, section title, paragraph, link, ...).

Informatics jargon. Web searches, which are allowed for all proofs, are mainly used for this part of the contest. Contestants have the chance to discover the meaning of words or expressions they already heard but whose meaning they ignore (often also because they are not translated into Italian), to dissolve false ideas, and to correct the improper use of some technical expressions. Moreover, the intent is to tell some stories or anecdotes about the history of informatics, computer scientists, and informatics practitioners.

In 2009 the teams had to fill in the blanks in some short stories. Missing keywords were for instance Alan Turing, artificial intelligence, CamelCase, spam, or Linux. In 2010 we used a bad translator from English to Italian to produce some incomprehensible and funny texts about files, hardware components, or operations on computers. We exploited the double meaning of words – for instance, *file* is also the English word denoting "a metal tool with a rough surface for cutting or shaping hard substances or for making them smooth"; or the word *folder* can be seen as formed by the verb "to fold" plus a suffix "-er", and hence can be translated with the meaning of a "person who bends something, especially paper or cloth, so that one part lies on top of another part"! In 2011 we joked with geek humour, inspired by those T-shirt with geeky sentences written on front: "to understand recursion first one must understand recursion", "there are only 10 types of people in the world: those who understand binary and those who don't", "there's no place like 127.0.0.1", "2B | [^B]{2}". Each sentence had to be related to an image: a fractal picture for recursion, an image representing the conversion of a decimal number into binary, a warmy home for the localhost IP,

[2] It should be noticed that the tool implemented to support teams in testing their solution actually built the presentation using LaTeX-Beamer instead of PowerPoint since this gives much more control on structure and styles!

a portrait of Shakespeare for the "to be or not to be" regular expression. Clearly, to guess the right matchings, teams had to first understand the general meaning of the sentences or, at least, their context.

4 Participants Feedback

We are currently setting up collaborations with teachers to be able to collect a formal feedback from the participants and correlate it with the participants' performance and backgrounds. We already monitored some of the web pages and the student newspapers the pupils wrote after the participation. They describe the experience as fun and challenging. A common remark is that some of the issues, for instance those about jargon, are quite strange and unexpected; probably their comments are due to the fact that they did not associate with computer science many of the terms and expressions we proposed (like *regular expressions* or *camel case* or *root*). The choice of organizing the game around teams has a positive side effect: in order to be able to participate, pupils more focused on computers solicit friends with lesser interest, and the teams are indeed heterogeneously composed.

A strong positive feedback also comes from the teachers. In particular, they welcome the availability of the booklets we prepare: whenever they want to discuss informatics in their classes, they usually have to choose between specialized literature (mostly out of reach for a high school audience in a non technical environment) and ICT/business-oriented publications. The descriptions of the Kangourou challenges, instead, provide a way to introduce a topic and how it relates to applimatics and ICT and the references provide pointers to further studies. We are also considering the preparation of other types of support that are more specific to class work.

5 Conclusions

We firmly believe that informatics is a scientific discipline with an important educational value and sufficiently basic to be taught as a fundamental formative subject. It is also our belief that computing is indeed fun and that, by playing and working in team, pupils can discover some of the most important aspects of this discipline. We used these characteristics to organize the *Kangourou of Informatics*, a two rounds national game-contest. In the qualifying round we can attract a wide audience to the basic issues of informatics. In the final round we can engage pupils in more challenging tasks and offer refresher courses and laboratories to their teachers. In this way, pupils can experience what informatics is, and go beyond ICT, which is the usual approach of schools to this subject.

References

1. Antonitsch, P.K., Grossmann, A., Micheuz, P.: Beaver, Kangaroo and classroom situations: A promising symbiosis. In: Hromkovič, J., Královič, R., Vahrenhold, J. (eds.) ISSEP 2010. LNCS, vol. 5941, pp. 16–31. Springer, Heidelberg (2010), http://www.issep2010.org/proceedings_of_short_communications.pdf

2. Barr, J., Cooper, S., Goldweber, M., Walker, H.: What everyone needs to know about computation. In: Lewandowski, G., Wolfman, S.A., Cortina, T.J., Walker, E.L. (eds.) Proc. of the 41st ACM Technical Symposium on Computer Science Education, SIGCSE 2010, Milwaukee, Wisconsin, USA, pp. 127–128. ACM, New York (2010)
3. Ben-Ari, M.: Constructivism in computer science education. In: Lewis, J., Prey, J., Joyce, D., Impagliazzo, J. (eds.) Proc. of the 29th SIGCSE Technical Symposium on Computer Science Education, Atlanta, Georgia, USA, pp. 257–261. ACM, New York (1998), http://doi.acm.org/10.1145/273133.274308
4. Brinda, T., Puhlmann, H., Schulte, C.: Bridging ICT and CS: educational standards for computer science in lower secondary education. In: Brézillon, P., Russell, I., Labat, J.M. (eds.) Proc. of the 14th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education, ITiCSE 2009, Paris, France, pp. 288–292. ACM, New York (2009), http://doi.acm.org/10.1145/1562877.1562965
5. Carnegie Mellon University: Alice website, http://www.alice.org
6. Dagiene, V.: Teaching information technology in general education: Challenges and perspectives. In: Mittermeir, R.T. (ed.) ISSEP 2005. LNCS, vol. 3422, pp. 53–64. Springer, Heidelberg (2005), http://dx.doi.org/10.1007/978-3-540-69924-8_27
7. Dagiene, V.: Information technology contests – Introduction to computer science in an attractive way. Informatics in Education 5(1), 37–46 (2006), http://www.mii.lt/informatics_in_education/htm/INFE069.htm
8. Dagiene, V.: Sustaining informatics education by contests. In: Hromkovič, J., Královič, R., Vahrenhold, J. (eds.) ISSEP 2010. LNCS, vol. 5941, pp. 1–12. Springer, Heidelberg (2010), http://dx.doi.org/10.1007/978-3-642-11376-5_1
9. Dagiene, V., Futschek, G.: Bebras international contest on informatics and computer literacy: Criteria for good tasks. In: Mittermeir, R.T., Sysło, M.M. (eds.) ISSEP 2008. LNCS, vol. 5090, pp. 19–30. Springer, Heidelberg (2008), http://dx.doi.org/10.1007/978-3-540-69924-8_2
10. Squeak Etoys website, http://www.squeakland.org/
11. Hromkovic, J.: Contributing to general education by teaching informatics. In: Mittermeir, R.T. (ed.) ISSEP 2006. LNCS, vol. 4226, pp. 25–37. Springer, Heidelberg (2006), http://dx.doi.org/10.1007/11915355_3
12. Lissoni, A., Lonati, V., Monga, M., Morpurgo, A., Torelli, M.: Working for a leap in the general perception of computing. In: Cortesi, A., Luccio, F. (eds.) Proceedings of Informatics Education Europe III. IFIP, pp. 134–139. ACM, New York (2008), http://www.dsi.unive.it/IEEIII/atti/PROCEEDINGS_IEEIII08.pdf
13. Mirolo, C.: Quale informatica nella scuola (2003) (in Italian), http://nid.dimi.uniud.it/pages/materials/discussion/educazione.pdf
14. MIT Media Lab: Scratch website, http://scratch.mit.edu/
15. Papert, S.: Teaching children to be mathematicians vs. teaching about mathematics. Artificial Intelligence Memo AIM-249, MIT (1971)
16. Verhoeff, T.: The role of competitions in education (1997), http://www.win.tue.nl/~wstomv/publications/competit.pdf, presented at Future World: Educating for the 21st Century, a conference and exhibition at IOI 1997
17. Verhoeff, T.: The IOI is (not) a science olympiad. Informatics in Education 5(1), 147–159 (2006), http://www.mii.lt/informatics_in_education/htm/INFE078.htm