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]
Mauro Torelli[1]

[1]Dip. di Scienze dell'informazione
Università degli Studi di Milano, Italy

[2]Dip. di Informatica e Comunicazione
Università degli Studi di Milano, Italy

Google

Informatics is fun...

...but only few discover it in schools!

- informatics == computer/applications literacy
- learn it because it helps you in finding a job
- the fun is normally associated to specific entertaining uses of computers (games, social networks, etc), not the discipline and its challenges

Instead, we know it's fun:

> *"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." [Alan Perlis]*

How to show it to young people?

Why misperceived?

Main problem: What is the role of computers (and applications) in informatics (computer science)?

"Computer science is no more about computers than astronomy is about telescopes*."* [Dijkstra, 1986]

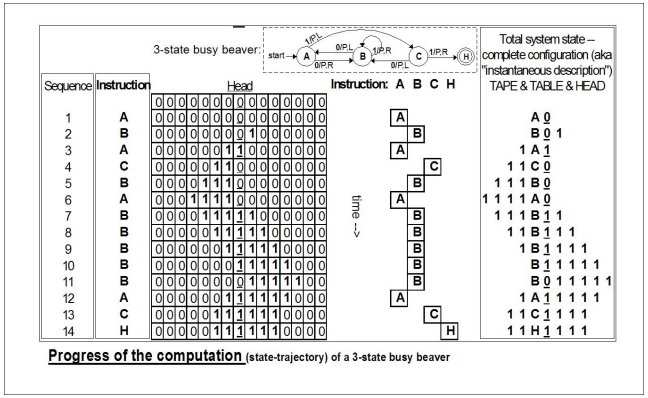
"The term computer science is as descriptive as the etymology of the word geometry*"* [Abelson, 1987]

Among experts the answer is rather clear: computer science is not (or not only) the science of computers, but rather the science of computing (for this reason the name Informatics is preferred, at least in Europe).

Computer Science

In fact, informatics is a multi-faceted discipline seen in three radically different ways when taught in school:

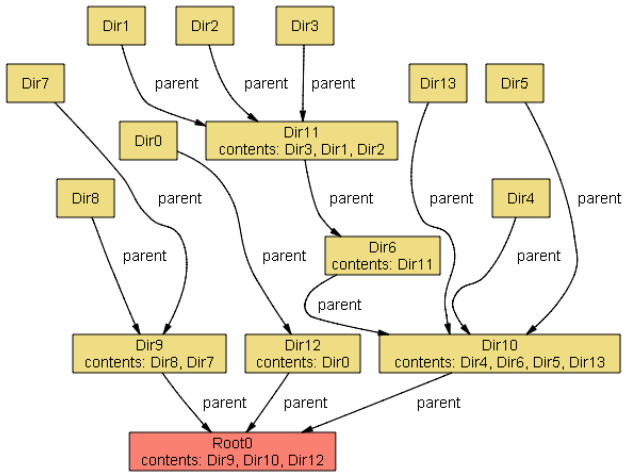
1. as a **science**, with its own peculiar approach to problem solving



Computer Science

In fact, informatics is a multi-faceted discipline seen in three radically different ways when taught in school:

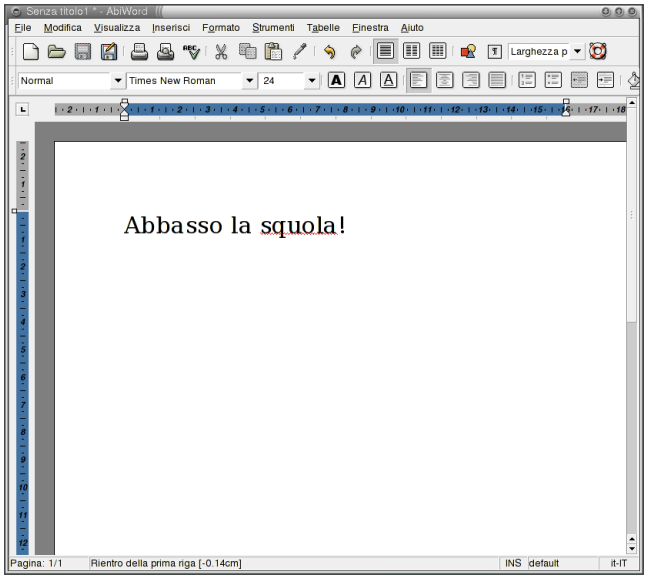
1. as a **science**, with its own peculiar approach to problem solving
2. as a **technology**, producing hardware and software tools



Computer Science

In fact, informatics is a multi-faceted discipline seen in three radically different ways when taught in school:

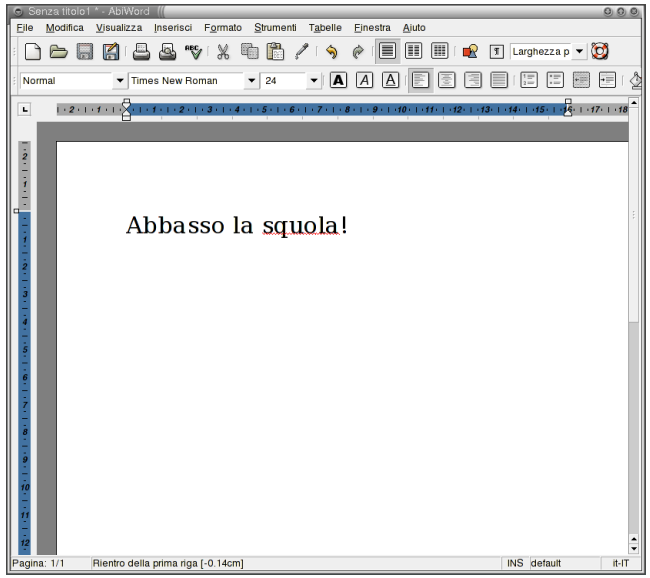
1. as a **science**, with its own peculiar approach to problem solving
2. as a **technology**, producing hardware and software tools
3. as an **instrument**, to work on problems arising in all contexts



Computer Science

In fact, informatics is a multi-faceted discipline seen in three radically different ways when taught in school:

1. as a **science**, with its own peculiar approach to problem solving
2. as a **technology**, producing hardware and software tools
3. as an **instrument**, to work on problems arising in all contexts



But the general public has a reductive perception of informatics as the mere ability to master a set of applications or communication tools!

By neglecting the ‘science and technology’ facet we lose the most intellectual fertile part of informatics!

What can be done?

Shift the focus

A cultural battle is due: Expose children to...

...computers

...computing

We must show the computational core of informatics to children!

A formative subject

We actually believe informatics is a very formative discipline, not just for a specialist audience.

- abstraction
- algorithmic thinking
- structured problem solving
- computation with constraints on the resources (time and space)
- learn about how we learn (S. Papert)

Promoting through contests: The Kangourou

In 2008 we were contacted by "Kangourou Italia" who organized the Kangourou of Mathematics with another department of our University.
Kangourous are a lot of fun. . . why not organize one focused on Informatics?

- The Kangourou has received many awards for its contribution to the promotion of mathematics among young people.
- K. of Mathematics engages about 50'000 primary and secondary school pupils in Italy
- The contest is the opportunity for distributing pleasant documentation to pupils and teachers.

Kangourou vs. Olympiads

Olympiads *citius, altius, fortius...* the quest for excellence; pupils are engaged in a rather specialized training

Kangourou Promote the interest in the discipline among young people (and teachers) even if informatics is not their major topic

K. doesn't assume any specialized knowledge (programming languages, data structures, etc.). In fact, we have goals similar to Beaver's (that we didn't know when we started).

The goals

The game-contest should stimulate pupils with regard to basic concepts:

- the description of unambiguous computing processes (*syntax and semantics*);
- the composition of complex objects from simpler elements (*abstraction and composition*);
- the analysis of the properties of computing processes (*complexity and computability*);
- the knowledge necessary not to uncritically accept the overwhelming technical jargon.

Caveats

We identified three main obstacles:

1. the abstract nature of computing
2. the use of specialized language and terms
3. the risk of disappointment, since pupils expect computers

The Kangourou of informatics

Our assumptions

- No specific technical knowledge about programming languages or other formalisms for the representation of algorithms
- Logical, problem solving, and information coding abilities, as stemming from their mathematical background
- We expect some terminology is recognized as common jargon (bit, password, etc.) but not necessarily understood in its proper meaning

Peculiarities of KoI

- A team (4 pupils) contest
- Two categories: "Medie" (grades 6–8) and "Biennio" (grades 9–10)
- Two phases: a preliminary game (played in the schools) followed by a national final for the best 24 teams (max 1 per school)
- Multiple choice and interactive questions for the preliminary phase with the use of an ad-hoc software
- Open questions for the final.
- We produce and distribute a booklet with commented solutions to pupils and teachers (key to improve informatics perception from year to year)

Qualifying round

- carried out on-line from schools, under the supervision of teachers
- Ad hoc software: keeps the time, collects the answers and sends them to the server for evaluation
- Participants may use the Web or other applications, if they want: the focus is not on the tools, but the use of a computer application is important to keep the link explicit (yes, it is a computer science game!)
- Challenges with different difficulty: easy, medium, difficult (nobody should be able to do everything, but everyone should solve something and get captured by most of the challenges)

Example

Riccardo has two lists of soccer teams, he wants to know which teams are in the first but not in the second. Riccardo may combine (by pipe and filters, we might say) 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.

Two versions. Medie: identify a correct solution. Biennio: build a correct solution.

Example

Medie: identify the correct solution.

Squadre di calcio (3 punti)

Il piccolo Riccardo ha raccolto i nomi delle squadre di calcio che hanno partecipato ad almeno uno dei campionati passati e li ha messi nella lista "SquadreVecchie".
Poi ha messo nella lista "Squadre2011" l'elenco delle squadre partecipanti al campionato 2011.

Ora Riccardo vorrebbe sapere quali sono le squadre che partecipano per la prima volta, usando tre programmi di cui dispone il suo PC:

- un programma che attacca due o più liste una dopo l'altra;
- un programma che elenca gli elementi ripetuti in una lista;
- un programma che elenca gli elementi unici in una lista.

Ogni volta che un programma è eseguito, produce come risultato una nuova lista, senza modificare quella o quelle su cui ha operato.
In che modo Riccardo può utilizzare i programmi per ottenere il risultato voluto?

- E' impossibile
- Attacca "Squadre2011" e "SquadreVecchie"; trova nel risultato gli elementi ripetuti; li attacca a "Squadre2011"; elenca gli elementi unici di questa nuova lista
- Attacca "Squadre2011" e "SquadreVecchie"; trova nel risultato gli elementi ripetuti; li attacca a "SquadreVecchie"; elenca gli elementi ripetuti di questa nuova lista
- Attacca "Squadre2011" e "SquadreVecchie"; trova nel risultato gli elementi ripetuti; li attacca a "SquadreVecchie"; elenca gli elementi unici di questa nuova lista
- Attacca "Squadre2011" e "SquadreVecchie"; trova nel risultato gli elementi unici; li attacca a "SquadreVecchie"; elenca gli elementi unici di questa nuova lista

Example

Biennio: build a correct solution.

Squadre di calcio (max 7 punti)

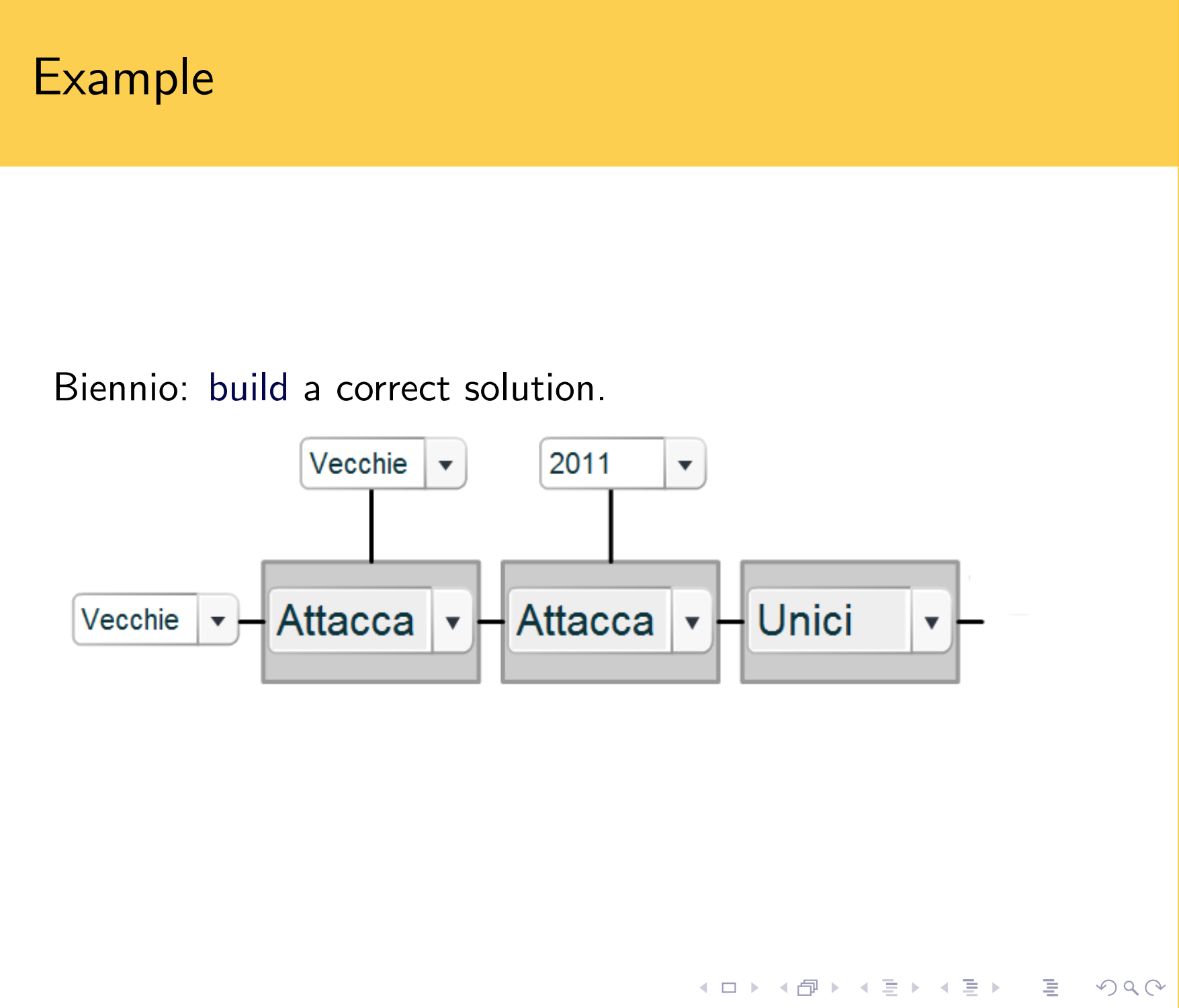
Il piccolo Riccardo ha raccolto i nomi delle squadre di calcio che hanno partecipato ad almeno uno dei campionati passati e li ha messi nella lista SquadreVecchie. Poi ha messo nella lista Squadre2011 l'elenco delle squadre partecipanti al campionato 2011.
Ora Riccardo vorrebbe sapere quali sono le squadre che partecipano per la prima volta, usando tre programmi di cui dispone il suo PC:

- attacca, che attacca due liste una dopo l'altra;
- ripetuti, che elenca gli elementi ripetuti in una lista;
- unici, che elenca gli elementi unici in una lista.

Ogni volta che un programma è eseguito, produce come risultato una nuova lista, senza modificare quella o quelle su cui ha operato.
In che modo Riccardo può utilizzare i programmi per ottenere il risultato voluto? Collega i blocchi corrispondenti ai programmi in modo da ottenere la lista desiderata. Dati due blocchi collegati, il risultato del programma nel blocco a sinistra verrà usato dal programma nel blocco di destra.

2011

Example



Biennio: build a correct solution.

Another example

Geomag (max 5 punti)

Gli allievi del dottor Kang si sfidano in un gioco di intelligenza. Compongono con il Geomag la costruzione di figura 1.
Lo scopo del gioco è ottenere tre quadrati come in figura 2 utilizzando i bastoncini e le biglie a disposizione.
Vince chi compie il minor numero di mosse, contando come mossa ogni bastoncino spostato. Lo spostamento delle biglie non conta invece come mossa.

Meno modifiche farete più punti otterrete!

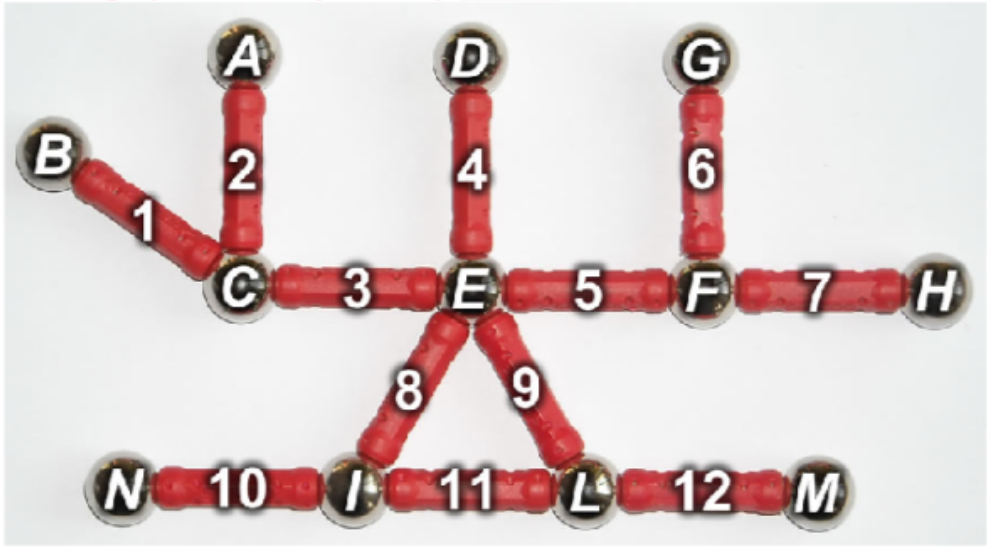


Figura 1

Mossa	Bastoncino da spostare	In modo che colleghi	
		La biglia	Alla biglia
1			
2			
3			
4			
5			
6			

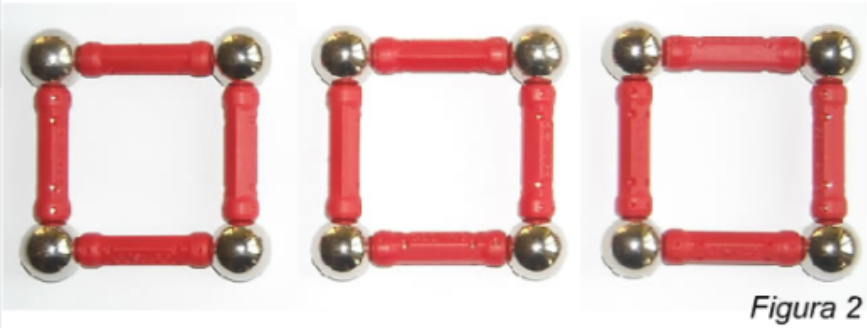
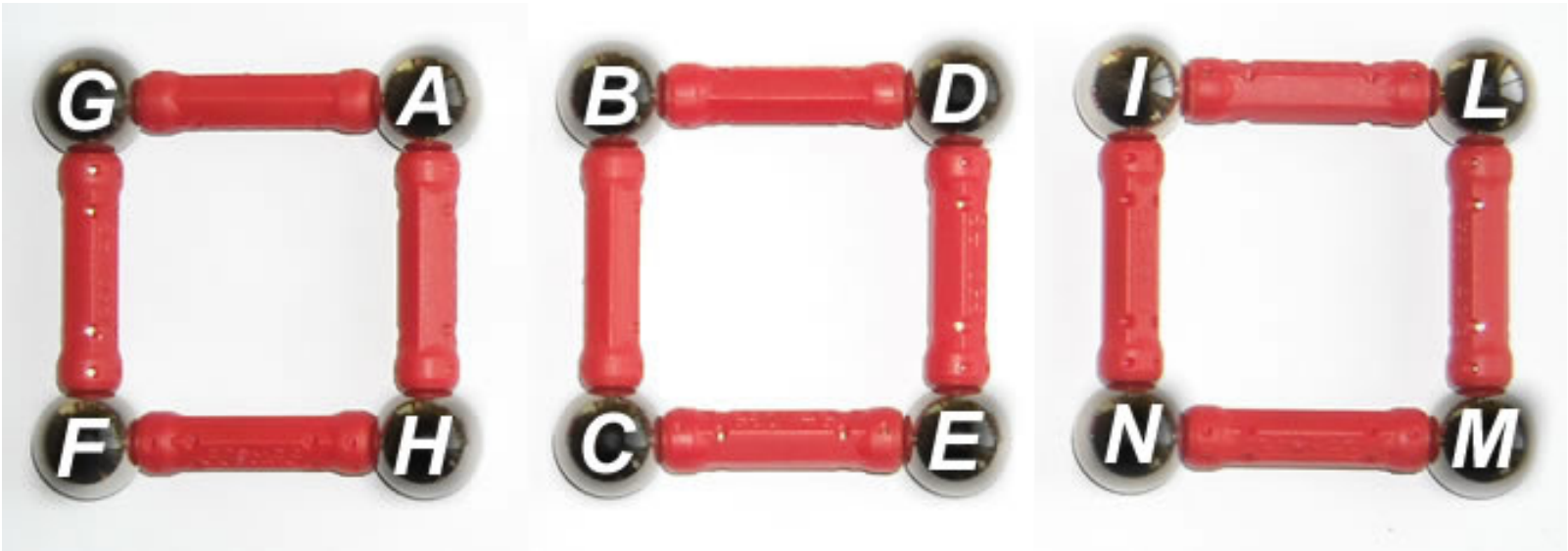


Figura 2

Another example



Final round

- Final round in Mirabilandia (an amusement park near Rimini)
- $12 + 12$ best qualified teams (at most one per school)
- Lessons and labs for teachers (while pupils play the game)
- Paper puzzles **and** computer based tasks
- Computers are available in "time sharing": every PC is shared between two teams, teams are automatically logged out when a time slice is elapsed

Mirabilandia

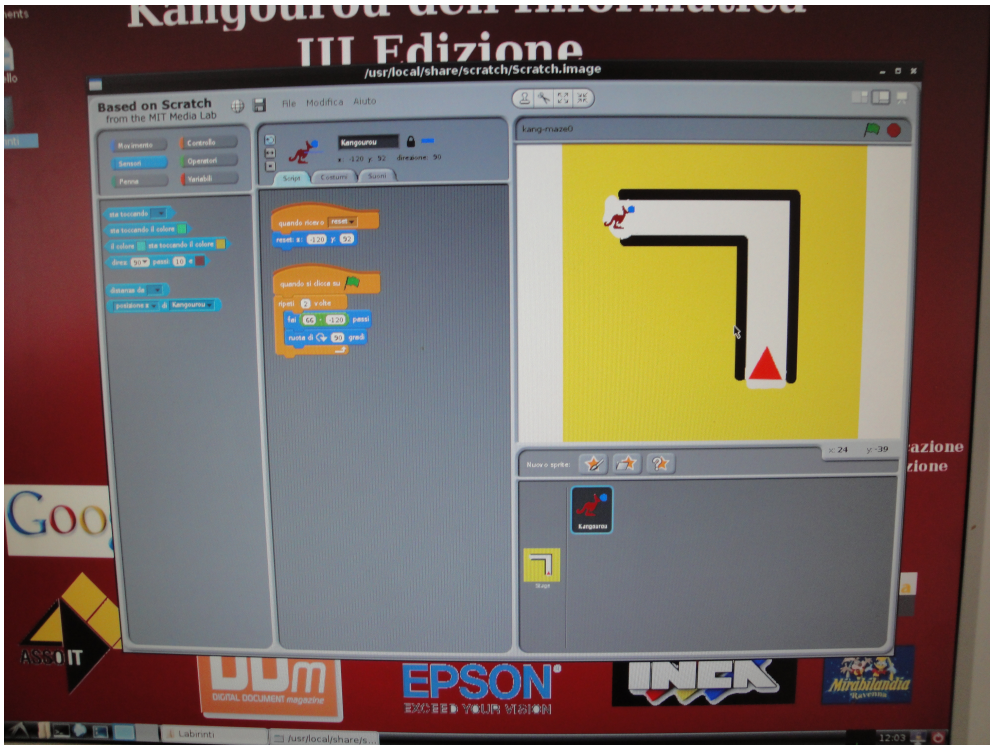
The 2011 finalists

Themes

Challenges (mainly with open solutions) focus on

- Problem solving and “programming”
- Text description and structuring
- Informatics jargon

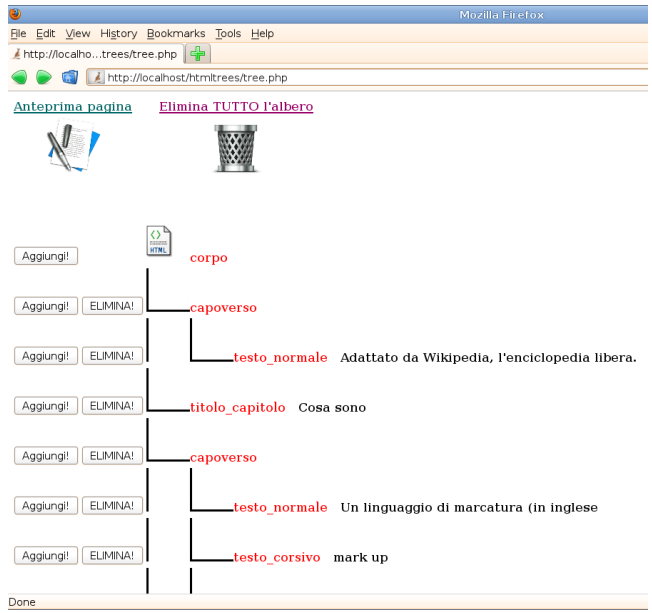
Programming example



Reflect on text description and structure

Lorem ipsum dolor sit amet, **consectetur adipisici elit**, sed eiusmod tempor incidunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, *quis nostrud exercitation* ullamco laboris nisi ut aliquid ex ea commodi consequat.

- capoverso:
  - testo_normale: Lorem ipsum dolor sit amet,
  - testo_grassetto: consectetuer adipiscing elit
  - testo_normale: sed eiusmod tempor incidunt ut labore et dolore magna aliqua. Ut enim ad minim veniam,
  - testo_corsivo: quis nostrud exercitation
  - testo_normale: ullamco laboris nisi ut aliquid ex ea commodi consequat.



Jargon


There's no place like 127.0.0.1
I am root!
If you see me laughing, you'd better have a backup
Friends don't let friends use Comic Sans.
RECURSION, ONE MUST FIRST UNDERSTAND RECURSION, ONE MUST FIRST UNDERSTAND
There are only 10 types of people in the world: Those who understand binary and those who don't.


1/2
SHAKESPEARE

The participants

Feedback

- The comments we collected from participants describe the experience as fun and challenging
- 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
- 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 and a good ability in avoiding downtime
- Positive feedback from the teachers: they also welcome the availability of the booklets, a good alternative to specialized literature (mostly out of reach) or commonplace ICT/business-oriented publications

Ongoing work

- Collaborating with teachers to understand how informatics fits in non vocational curricula
- Engage pupils in computing through "algomotricity": physical activities that simulate computations
- Collect more detailed statistics on pupils background to correlate it with performance
- Join the Beaver international effort

Conclusions

We believe we are on the right path:

- The number of participants is increasing
- Schools tend to participate again
- The feedback is mostly positive both from teachers and pupils
- It is still not easy to win many teachers' fear of inadequacy

Acknowledgments

Kangourou of Informatics is made possible by the work of many:

- Kangourou Italia 
- The organizing team: Fabrizio Carimati, Angelo Lissoni, Violetta Lonati, Mattia Monga, Anna Morpurgo, Roberta Raimondi, Nicola Rebagliati, Lorenzo Repetto, Mauro Torelli
- Several master and phd students (too many to list!) helped in all the phases
- Several sponsors supported the activities: Mirabilandia, Google, Microsoft, IBM, ...