Computer-assisted Teaching in Class Situation: a High-school Math Lab on Vectors

Maud Marchal
TIMC/IMAG
Maud.Marchal@imag.fr

Peggy Provent
INSERM/UJF
Peggy.Provent@ujf-grenoble.fr

Frederic Ruyer
LAMA/Université de Savoie
Frederic.Ruyer@univ-savoie.fr

Pirouz Djoharian
French National Education
Pirouz.Djoharian@free.fr

Fabrice Neyret
GRAVIR/IMAG-INRIA
Fabrice.Neyret@imag.fr mobinet@imag.fr

Abstract

This paper presents our design and experiment of a computer-assisted class laboratory on vectors in high-school. Our main goal is to improve the acquisition on notions by all pupils, from advanced level to remedial level. Our secondary goal is to improve the motivation of pupils toward maths and sciences.

Our approach consists in making the pupils *practice* the notions included in a given set of pedagogic objectives, and to do it through real world-inspired problems. We draw on *MobiNet*, a platform that simulate programmable mobiles on screen and through the network. Our class laboratory consists of a set of pre-programmed interactive simulations, allowing the pupils to react on their errors and to validate each exercise in an autonomous way. This class laboratory was conducted in the same conditions and constraints than an ordinary math lab: same duration, and no preparation of the pupils (but the ordinary math course). Still, the autonomy of the exercises and the network ability would also allows the use with no teacher on place. In this paper, we describe our vector class laboratory experiment: objectives, design, conduction, and evaluation.

Keywords: computer-assisted teaching, pedagogy, maths class laboratory, simulation

1 Introduction

1.1 MobiNet context

Learning maths and sciences in high-school is tedious for pupils: in France, the official Academic Program is dense, matters are abstract and are loosely connected to the real world, learning is mainly passive.

Conversely, numerous softwares exist to play with maths and sciences. But making use of them in class in real conditions and real constraints is not easy. Still, manipulating notions instead of passively learning them is known to be the best way for understanding. Moreover, this helps pupils who are not comfortable with abstraction (who might be rejected by the scholar system), as well as pupils with an advanced level (who are sometimes not aware that mathematical notions are tools).

The main problem of such an approach is the time it requires, in addition to the design of scenarios ensuring that the whole strict *of-*

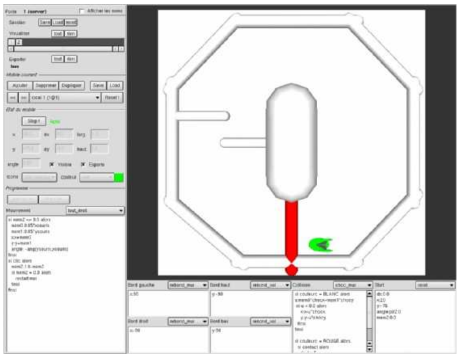


Figure 1: The MobiNet worksheet.

ficial set of notions is acquired by the pupils. Moreover, the official Academic Program contains numerous matters and subjects to be acquired in a given time having a given number of teachers: if a new pedagogic tool is to be used in class, it must fit the available *time slot for a lesson* (formation to the tool included, if required). Our main goal is to implement such an approach while insuring the constraints mentioned above.

Another issue is the fact that fewer and fewer pupils are studying sciences after high-school. We experienced [1] [1, 2] that numerous pupils – even if they have understood the theoretical notions – do not figure out that their maths and physics courses are related to the real-world, i.e. the modeling of Nature as well as the engineering tasks in industry. As an answer, our secondary goal is to motivate the purpose of notions by building real-world inspired problems.

In this paper, we present the experiment we conducted in high-school through a math class laboratory about vectors, using the *MobiNet* platform [3]. This simulation software allows to program and tune the behavior of mobiles on screen and through the network (see figure 1). Its principles are reminded in Section 1.2 and Figure 6. In Section 2 we describe our target pedagogic context and the list of requirements. We detail in Section 3 our pedagogic scenario and how to adapt it on MobiNet. The progress of the class laboratory is described in Section 4 (in principles as well as in its live version). We analyze the results in Section 5.

1.2 *MobiNet*

MobiNet is a software allowing the manipulation of graphical objects called *mobiles*. The user can interactively create such objects, specify their state variables (i.e. attributes) and their behavior programs. A dedicated programming language[2] is provided to the user to describe the evolution of the state variables of the mobiles and the mobiles interactions. The behavior of the mobiles can be interactively modified through the graphical interface or through formulas entered in the behavior control boxes (a quick overview of MobiNet is provided in Appendix A).

Complex applications can be built, saved and loaded: e.g., physical simulations, computer-assisted maths figures, games, etc. Interaction elements such as sliders, handles or buttons are easily implemented as mobiles. Thus, the teacher can use MobiNet as an easy programmable simulation software, or use existing exercises sessions (which he can modify). In such a case, the pupils simply use the simulation without creating or programming mobiles. This is the modality we followed in our class laboratory.

An interesting characteristic of MobiNet is that it can export and share mobiles through a network. So, a user can interact with other connected sessions. Collaborative exercises can be built with several machines and can make an exercise more attractive for pupils. Since the communication protocol is efficient and robust, network can be the Internet. This also allows a distant teacher to watch each of the pupils screen. Video-projection of any screen for a master station is also ensured this way.

2 Context and List of Requirements

2.1 *Pedagogic context*

We have chosen a mathematic class in the French equivalent of the 10^{th} grade (two years before the final high-school exam), since impor-

[1] in the context of the biannual "Engineerings Weeks" of the Engineering School federation INPG (Institut National Polytechnique de Grenoble)

[2] The language reference is available on `http://www-evasion.imag.fr/mobinet/` (in French and English).

tant carrier orientation choices are done at this level.

The class laboratory took place in the computer room with half-classes, a week after the lesson. One hour was available for each group (approximatively 20 pupils each). This maximum duration constraint is strict.

We chose the subject in close collaboration with the teachers, taking into account the topics which are difficult for the pupils, the estimated appropriateness of MobiNet, the lessons schedule and the possible dates along the year. Geometry, trigonometry, functions, and possibly statistics where identified as adapted fields within the official10^{th} grad program. Others constraints made us converge onto the lesson on vectors. As pupils face to difficulties with space representation of the manipulated objects, a graphical application like MobiNet is particularly well appropriate to give a representation of geometrical notions.

2.2 *Pedagogic objectives*

The vector lesson corresponds to three pedagogic objectives:

- Constructing a vector from two points (with coordinates);
- Adding two vectors;
- Multiplying a vector by a scalar.

The pupils should also be able to subtract two vectors, to combine the operations and to compute vector norms. The pupils have learned coordinates, but not yet trigonometry.

We aim at producing an exercise session training and validating the understanding of these notions. To ease the acquisition, we aim at introducing these notions progressively. Our approach is to make pupils manipulating practical situations (preferably real-world inspired) requiring these notions. Still, we have to make sure that random manipulation cannot leads to validation. Conversely, we have to avoid excessive manipulation with no added-value. Given the short available time only interactive manipulations and (reasonable) numbers feeding can be considered, and not complex scenario implying creating and programming new mobiles.

3 Exercises design

3.1 *The chosen scenario*

We have chosen the theme of a boat race figured from top view (see Figure 2):

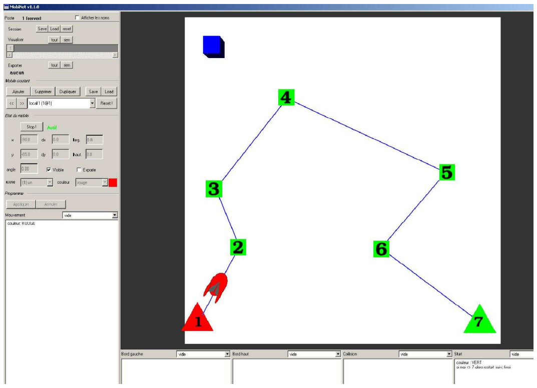


Figure 2: Target result.

Numbered buoys are placed to trace the race map. The pupils objective is to make the boat following the trajectory from the initial buoy to the final buoy by crossing all the buoys in the good number orders. The trajectory is represented on the screen as successive segments linking two buoys.

The pupil motivation is to control the trajectory of its boat like in the corresponding real situation of a boat race. A sailor drives namely its boat between different buoys by heading for the following buoy located in a given distance. Here, the control is done by providing the pilot-robot with the vector direction to head at at each step, taking into account possible water currents. The velocity of the boat is assumed to be constant.

We propose three exercises which introduce progressively the three notions:

- In the first, the pupils simply have to construct for each step the vector to follow in order to join the next buoy, by subtracting the coordinates of the buoys.

- In the second, a global current deviates trajectories by a constant amount, so that the heading direction differs from the trajectory direction. The pupils have to take this deviation into account to define the heading vector by subtracting the deviation vector to the direction vector.

- In the third exercise, we consider a more realistic model of deviation which is proportional to the path length. So, the pupils have to subtract the deviation vector weighted by the path length, to the direction vector.

Figures 2 and 3 illustrate the solution the pupils have to obtain without and with currents (the arrows figure the heading direction entered by the pupils).

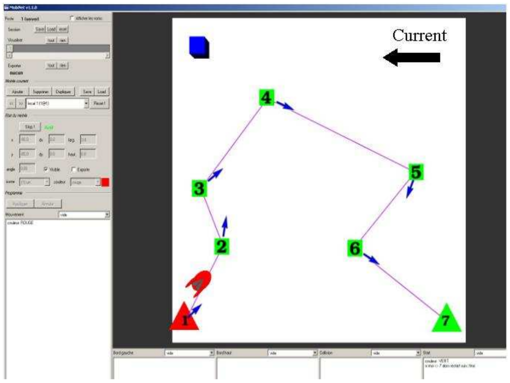


Figure 3: Target result with current.

3.2 The pupils job

Figures 4 illustrates the initial configuration the pupils start with. The pupils can read the coordinates of a given buoy in A by selecting its mobile number in B. After computing the heading vector for a given step of a given exercise, the pupils enter the vector coordinates in C at the beginning of the program ruling the corresponding trajectory segment (selected via B as well), under the form vx : ...vy : ... . The corresponding segment of trajectory is immediately updated on screen.

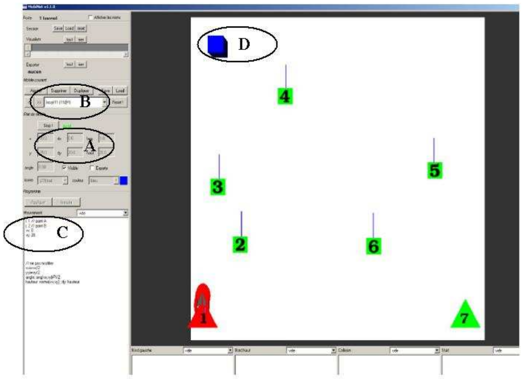


Figure 4: Initial configuration: the different parts of the interface are zoomed and detailed in Appendix A.

At any time the pupils can validate their construction by clicking on the button D. This makes the boat following the trajectory from the first buoy to the last or to stop on the first erroneous buoy (the color of validated buoys turns from red to green). Beyond checking errors, this precise validation guarantees that the trajectory was not obtained by chance or by iterative tuning.

Once the pupils have shown to the teacher that they succeeded, they can load the next exercise.

3.3 The programmer job

Programming a session in MobiNet consists of several parts: creating every necessary mobiles with default attributes (look, location), and programming their behavior (motion, or any change or reaction to events) in the appropriate program boxes using the MobiNet language (which syntax is very simple ^{2}).

Here, buoys are motionless mobiles which icons is chosen to figure their number. Their only action is to change color when triggered by the validator (i.e. the boat). This action is defined in the `restart` box (since stopping or restarting another mobile is the way mobiles can trigger each others in MobiNet, besides collisions). We chose seven buoys, i.e. six successive trajectory segments, as a good compromise between realistic complexity and amount of pupils work.

The current corresponds to a mobile figured by an arrow icon, which length tunes the intensity. To help other mobiles programs checking its value, its action consists in evaluating its Cartesian coordinates in its `dx` and `dy` state variables:

```
dx:  l*cos(angle)
dy:  l*sin(angle)
```

The segments joining the buoys (once their direction is completed by pupils) are figured by a bar icon. Since the icon handle is at the middle, a couple of program instructions is required to infer the right position and angle from the previous buoys and from the heading variables the pupils have to define at the beginning of the program. Namely, the program (defined in the motion box since it is permanently updatable) is:

```
vx:  <...> ; vy:  <...>
i:  <1st buoy number>
angle:  ang(vx,vy)
x:  xi + vx/2 ; y:  yi + vy/2
```

In case of constant current (defined by mobile `<c>`), we correct the heading to get the right trajectory, just before computing `angle, x` and `y`:

```
vx:  vx + dx <c>
vy:  vy + dy <c>
```

(spaces can be added anywhere in MobiNet).
For currents proportional to length, the correction is:

```
n = norm(vx,vy)
vx:  vx + n*dx <c>
vy:  vy + n*dy <c>
```

In case of currents we also display arrows to figure the headings provided by the pupils. For this, we create extra mobiles with arrow icons, unitary lengths, and action `angle:mem1<i>` where `<i>` is the corresponding segment number, and we make the trajectory segments saving in their `mem1` state variable the heading angle: `mem1:ang(vx,vy)` (inserted just before the correction which gets the real direction).

The most complex program is the validator, figured as a boat icon: a length coordinate s variates between 0 and the sum of the segment lengths. The current segment i is identified as $\sum_{j=1}^{i-1} l_j \leq s < \sum_{j=1}^{i} l_j$, and the location of the mobile along the current segment is linearly interpolated along the segment. The continuity of segments is checked when ends are passed, resulting either in triggering the valid buoys or stopping the boat (`if`...`endif` statements are available in MobiNet).

Lastly, a button (a mobile with a square icon) is used to allow the user to trigger (restart) the validator (i.e. to launch the boat simulation).

Sources of these sessions are available on the MobiNet web site[3].

4 The lab

4.1 Lab set-up and conduction

The lab took place in the computer room with a half class and last one hour (it was conducted twice). Pupils are in about 10 groups of two per computer. Besides MobiNet, they can use paper, and a real or a virtual calculator. A computer linked to a video projector is kept for explanations.

One assistant would suffice to introduce the lab, answer questions and verify that exercises are validated. In practice all the authors were present in order to observe the experiment and to be able to react to any potential problem. Thus, we shared the tasks and took more time introducing the experiment (to the risk of shortening the available time).

The lab session organizes as follows:

- In advance: Computer set-up (installation of MobiNet and of the exercises sessions; verification on each computer).
- Beginning of the session: Pupils log in and launch MobiNet. The teacher presents the experiment and the persons, whose introduce MobiNet.
- First part of the class laboratory: Brief presentation of the context of the lab session: How can we program a robot-pilot of a boat in order to realize a precise trajectory ? A booklet (available on the web site [3]) with the explanation and a brief reminder of the interface is provided. The rules are given: how to enter answers, requirement to validate then show it to the teacher before going to the next exercise. Pupils have to take notes in order to write a report after the lab. The possibility to save the sessions so that the pupils can keep a copy of their work is mentioned.
- Second part: Explanation of the first exercise. Individual help during the realization.
- Third part: Explanation of the second and the third part of the lab session when two thirds of the class have achieved the first exercise.
- Fourth part: Debriefing. Reminder of the report to be returned to the teacher, together with an opinion sheet.
- After the session: The pupils have to fill in a report justifying what they did and why, mentioning the laws they used. This was a base of our understanding analysis, completing the behavior observation during the lab.

[3] Sessions, lab booklet and complete report (in French) can be found at `http://www-evasion.imag.fr/moniteurs2005/`.

5 Results

5.1 Back-up of the session

The lesson on vectors was given the week before, without any exercise besides examples given in the course. One could not expect the pupils to have learned the lesson but globally the pupils were remembering the main facts. Still, several pupils asked some formulas to be reminded (especially six pupils which missed the course because of a trip out of school). The pupils had to log in their account. Time was lost due to computer problems or lost passwords. Precious time could be saved here by pre-logging and launching MobiNet on all computers before the pupils enter the room. The introduction of the experiment and of the tool lasted 20 minutes for the first group, which was too much. This was shortened for the second group.

We ended the first exercise after 20 minutes, and then explained the two next exercises. Only 20 minutes were remaining for the first group, so that only 3 pupils could do the third exercise (others could save their work). In the second group a third of the pupils could do it. In this last part, pupils could advance in the exercises asynchronously, which allowed to adapt to their skills.

According to the teacher, the content and the difficulties of the exercises were well adapted to the pupils and to the course but the lab was two long taking into account the installation and introduction time. Instead, we think we should decrease the installation and introduction time down to a minimal amount.

5.2 Teacher's observations concerning the pupils

Teacher's feelings about the pupils behaviors during the lab

The teacher did the following observations according to the usual behavior of various pupils:

The pupils showed a good motivation during the lab session. Several pupils which were usually passively waiting the end of the course were active in this lab. Some pupils usually abdicating get interested and were even more actively trying than "goods" pupils:the interactive manipulation on the computer and the practical aspect of the exercises gave them a chance to appropriate the notions.

The software mastery was pretty good. However, all the necessary formulas were not always known by the pupils. The presence of the teacher allowed individual help to the price of a delay in the exercises for these pupils. In addition, some pupils adopted a visual trial-and-error approach instead of an exact computing using to the formulas (strangely enough, these were the usually "good" pupils), which led them to a loss of time since the validator expected accurate results.

Teacher's analysis from the pupils reports

The pupils had to write a lab report after the class to re-explain the problems and to justify their answers. They did it seriously (in two to six pages, sometimes detailing the computation for every trajectory segments while only the principle was required).

From their answers, it occurs that they understood and applied correctly the formulas. Still, sign issues or confusion between heading vector vs trajectory vector sometimes leaded to complicated explanations (once, up to reversing the definition of vector coordinates from two points !).

More precise analysis and samples from pupils reports are available on the web site [3].

5.3 Pupils reactions

Although some pupils were afraid about computer manipulations, most of the pupils found MobiNet easier to use than they expected. The MobiNet interface was well appreciated and quickly mastered.

Some pupils with a bad level in mathematics were well involved. In contrary, some pupils which have a good mathematics level tried to turn around the expected work. Pupils globally enjoyed this lab and several asked for doing all their following lab sessions with MobiNet instead of the classical way.

An opinion sheet was provided to the pupils after the lab (see Figure 5) consisting of a set of questions requiring a mark in range 1..5 plus a room for free observations. Considering the average of marks (more details on the Figure), pupils found the lab interesting and useful and considered it helped them understanding the notions (3.7/5). They would like to do other ones with this tool (4.3/5). A third of the pupils considered it changed their mind about maths and physics!

6 Conclusion and future work

As we have shown, it is possible to produce computer-assisted class laboratory which are motivating and which helps pupils to better acquire knowledge, in the constrained framework of a class. As stated before, the autonomy of the exercises and the network ability of MobiNet would make it easy for a distant teacher to watch what the pupils are doing. Still, we would like to conduct deeper evaluations, including quantitative ones (e.g. relying on an exam given to two groups who had the same course with the same professor but different class laboratories).

MobiNet is probably not adapted to all possible topics, but we would like to investigate others math lessons. We would also like to investigate the physics courses, and possibly biology. Besides class exercises, MobiNet can also be used for virtual experiments (planetary systems, gas kinetics, nervous conduction, ...) and virtual setups (springs, ballistic trajectories, Galton experiment, biomechanics, ...). Examples are provided on the web site.

Out of the official program, we have also used MobiNet with 40 groups (half-classes per groups, 150 pupils a year) in half-day sessions (see [1]). The face motivation was to give an overview of programming, game making and CS carriers (in the context of INPG “Engineering week”, a biannual operation aiming at motivating high-school pupils to pursue scientific studies). During the 2.5 hours spent on the machines, the first hour is largely organized so to help pupils formalizing real situations into variables and equations and to manipulate mathematical notions as tools. (In the last hour, they program a pong game in network). Despite the fact that it does not correspond to a given lesson, this strengthens the notions of coordinates, vectors, amplitude, trigonometry, symmetries, etc, which were only theoretical in their mind (to the risk of piling up numerous knowledge not really mastered). Seeing the pupils happy and astonished to discover that their knowledge was really useful, allows them to produce something by themselves, and could even be used for fun, was a great satisfaction ! To some extent, the reaction of pupils after the math class laboratory on vectors was quite similar.

Acknowledgments

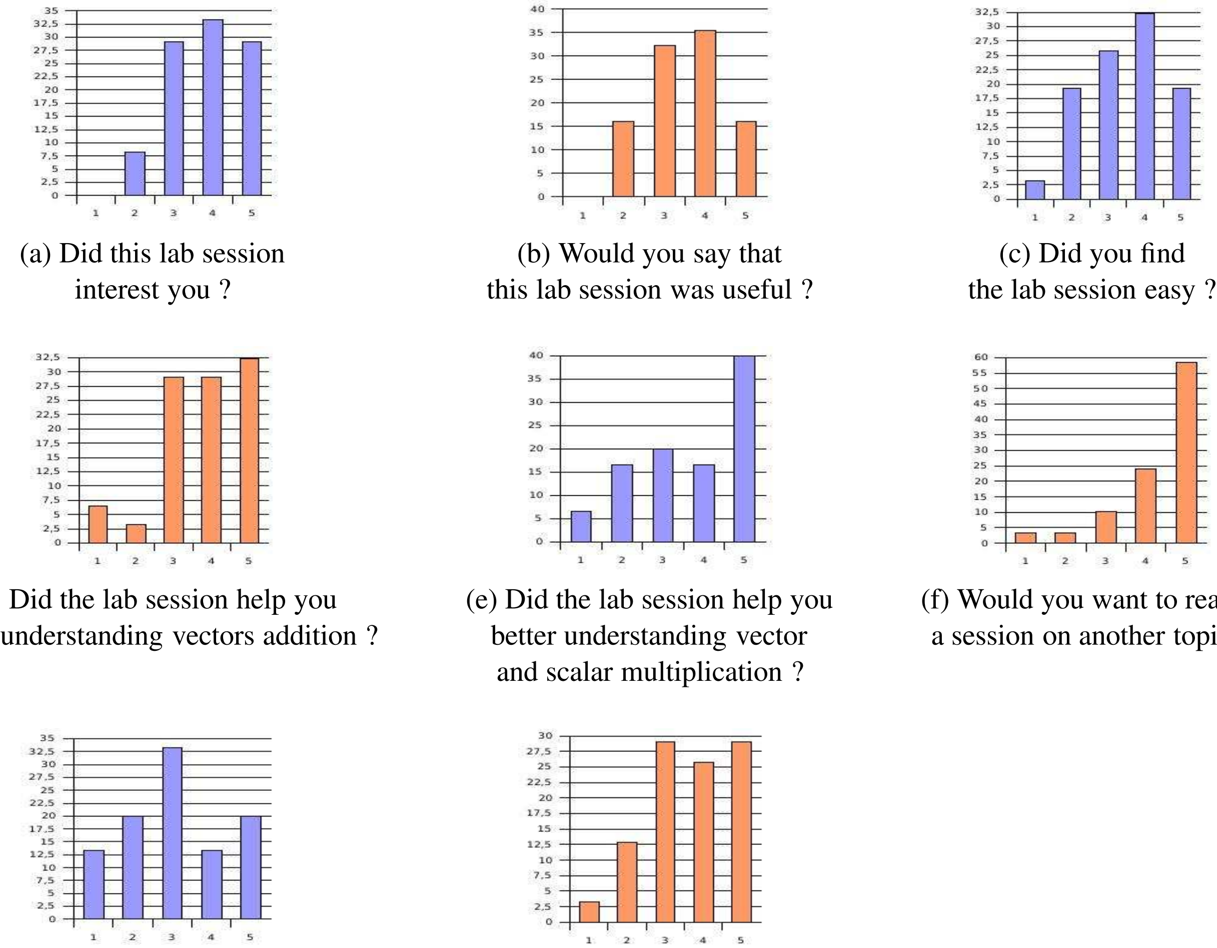
We wish to thank Michele Gandit, our contact at “Europole” high-school (Grenoble, France) and at the University Institute for Teachers Formation (IUFM-Grenoble) (the fourth author was a training teacher at “Europole”), Maryse Béguin and INPG (Institut National Polytechnique de Grenoble), who supported the MobiNet project since 2001, and CIES-Grenoble in the scope of which this pedagogic study was conducted as part of the formation of three of us (the three first authors). Thanks are also due to Joëlle Thollot for rereading the paper.

References

[1] Sylvain Lefebvre, Fabrice Neyret, Samuel Hornus, and Joëlle Thollot. MobiNet: a pedagogic platform for Computer Science, Maths and Physics (How to make students love Maths by programming video games). In *Eurographics - Education*. Eurographics, august 2004. Grenoble http://www-evasion.imag.fr/Publications/2004/LNHT04.

[2] Fabrice Neyret. MobiNet classes - analysis. In *Colloque Gout des Sciences:* http://www-evasion.imag.fr/mobinet/MOBINET-GoutDesSciences04.pdf, november 2004.

[3] MobiNet. Language, interface, tutorials, examples, papers: http://www-evasion.imag.fr/mobinet/. Technical report.



(a) Did this lab session interest you ?

(b) Would you say that this lab session was useful ?

(c) Did you find the lab session easy ?

(d) Did the lab session help you better understanding vectors addition ?

(e) Did the lab session help you better understanding vector and scalar multiplication ?

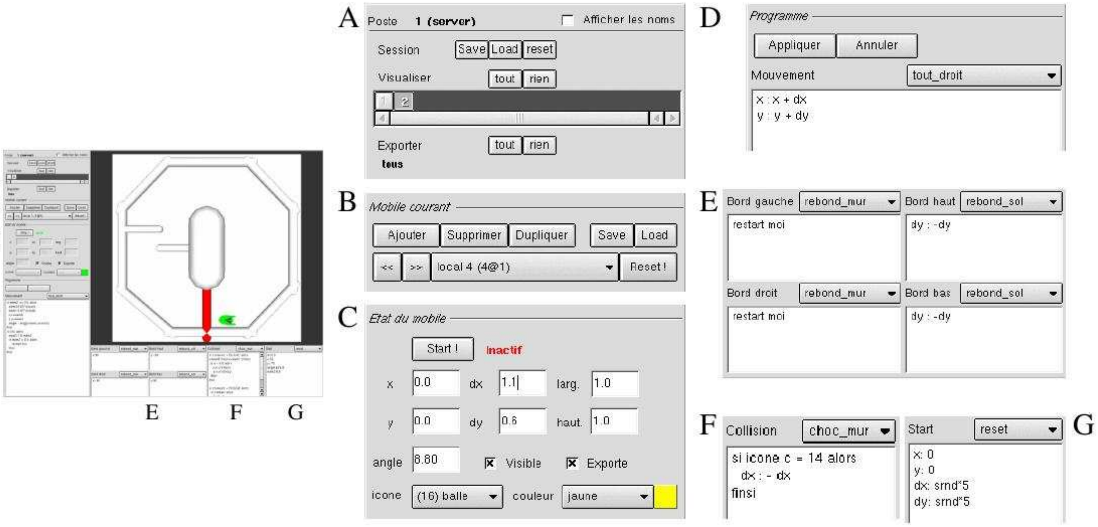
(f) Would you want to realize a session on another topic ?

(g) Did this session change your point of view on math and physics ?

(h) Do you think MobiNet in high school could be useful ?

Figure 5: Results of the pupils opinion evaluation. Columns corresponds to the percentage of answers for mark 1 (very low) to 5 (very high).

Appendix A: an overview of MobiNet



Mobile courant: local 4 (4@1)

Etat du mobile: inactif

x 0.0 dx 1.1 larg. 1.0

y 0.0 dy 0.6 haut. 1.0

angle 8.80 ☒ Visible ☒ Exporte

icone (16) balle couleur jaune 🟨

Mouvement: tout_droit

```
x : x + dx
y : y + dy
```

Bord gauche: rebond_mur

```
restart moi
```

Bord haut: rebond_sol

```
dy : -dy
```

Bord droit: rebond_mur

```
restart moi
```

Bord bas: rebond_sol

```
dy : -dy
```

Collision: choc_mur

```
si icone c = 14 alors
  dx : - dx
finsi
```

Start: reset

```
x: 0
y: 0
dx: srnd*5
dy: srnd*5
```

Figure 6: *Left*: General view of the screen. All the objects in the scene are *mobiles*, even when motionless. *Middle and right*: closeup on each interface elements controlling a given mobile.

A: The first row of button concerns loading and saving of the current session. The rest of this panel controls network sharing of mobiles. The user can choose whether he wants to interact with the exported mobiles of each MobiNet client running on the network (the user can also choose which of his mobiles he wants to export).

B: This panel contains all the user interface elements related to mobile creation. Mobiles can be added, deleted, duplicated, saved and loaded. The current mobile is chosen through a list or navigation buttons.

C: This panel displays the values of the state variables for the current mobile. These values can be directly edited, and are updated when the mobile is running. The state variables controlling the drawing of the mobile are its position `(x,y)`, its angle `(angle)`, its width and height `(haut, larg)`, its visible state `(visible)`, its shape `(icone)` and its color `(couleur)`. There exists supplemental all-purpose state variables: `dx, dy` (editable) and `mem1 ... mem4` (hidden). The export state (export) determines whether a mobile is accessible on the network. All the state variables can be modified either by program or by editing their values in this panel.

D: This panel contains the program for the motion of the mobile, which is executed at each time step (about 60 times per second). For an example entering `x : x + 1` would make the mobile go in straight line along the x axis. Attributes of other mobiles can be accessed (e.g., `x3`, or `x3@post4` through the network). The language (in English or French) allows classical algorithmic but loops. Its syntax is adapted to high-school notations.

E: These four text regions contain the programs executed when the mobile hits one of the screen borders. By default nothing is done (mobiles can escape the screen). Bouncing (for instance) can be programmed: assuming the state variable dx is used to store the motion direction (i.e., the motion equation is `x : x + dx`), a simple bouncing collision program could be `dx : -dx`.

F: This panel contains the program executed when a mobile collides with another mobile. Again, any behavior must be explicitly programmed: by default nothing happens (mobiles pass through each other).

G:This panel contains the program executed when a mobile starts (i.e. when the start button (see panel C) is pressed, or triggered by another mobile program). The start program is useful to initialize mobile state variables.