

MobiNet: a pedagogic platform for Computer Science, Maths and Physics (How to make students love Maths by programming video games)

Sylvain Lefebvre, Fabrice Neyret, Samuel Hornus, Joëlle Thollot

▶ **To cite this version:**

Sylvain Lefebvre, Fabrice Neyret, Samuel Hornus, Joëlle Thollot. MobiNet: a pedagogic platform for Computer Science, Maths and Physics (How to make students love Maths by programming video games). Eurographics - Education, Eurographics, 2004, Grenoble, France. inria-00510166

HAL Id: inria-00510166
https://inria.hal.science/inria-00510166

Submitted on 13 Oct 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

EUROGRAPHICS '0x / J.-J. Bourdin, S. Cunningham (Editors)

Volume 0 (2004), Number 0

MobiNet: a pedagogic platform for Computer Science, Maths and Physics
How to make students love Maths by programming video games

Sylvain Lefebvre, Fabrice Neyret, Samuel Hornus, Joëlle Thollot
GRAVIR/IMAG-INRIA Firstname.Name@imag.fr
http://www-imagis.imag.fr/mobinet/ mobinet@imag.fr

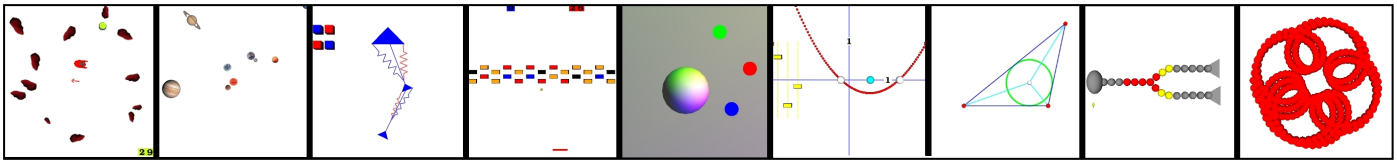


Figure 1: Different applications created with MobiNet. The behavior of each object (or *mobile*) is programmable through a very simple language. Mobiles can interact with each others. It is possible to build complex applications such as games or simulations of class notions.

Abstract

We developed the MobiNet (free) platform and tutorial sessions (tested on 16 batches of high school students) with the aim of offering students a new way of learning and understanding academic scientific subjects.
Our approach consists of letting the students ***manipulate*** *mathematical and physical notions as* ***tools*** *in order to solve concrete tasks, such as a solar system simulation or a video game. This makes students* ***formalize*** *a real-world problem, experiment by trial-and-error (error no longer means punishment), and gain an insight into the effect of equations and parameters. To allow this, our platform graphically simulates a world of* ***mobiles*** *where appearance, behavior and interaction (possibly through a network) are defined and programmed by the users, relying on an intuitive graphics interface and a dedicated programming language.*
This approach provides a strong motivation to students, and strengthens their understanding of theoretical notions seen in class (which is especially important for fundamental concepts used later in the curriculum including other academic subjects). Moreover, given the concern of academics and politicians in France about students no longer opting for scientific curriculums, we point out that our approach is closer to the real practice of engineers and researchers than traditional academic teaching. This helps students to figure out why taught notions are useful and how they can be related to professional activities (which they might consider for their future). This project has been tested in the context of the Engineering Week *organized by the Engineering school federation INPG (*Institut National Polytechnique de Grenoble*).*

1. Introduction

1.1. Principle

MobiNet is a program allowing the graphical simulation of a set of *mobile objects* interactively created by the user, who specifies *state variables* and *behavior programs*. It consists of a fullscreen window containing a large area for graphical display and various other areas used to specify the mobile state and programs. The purpose of the programs – which are usually very short and which rely on a dedicated language – is to modify the state variables in order to simulate trajectories, physical laws or more complex behaviors.

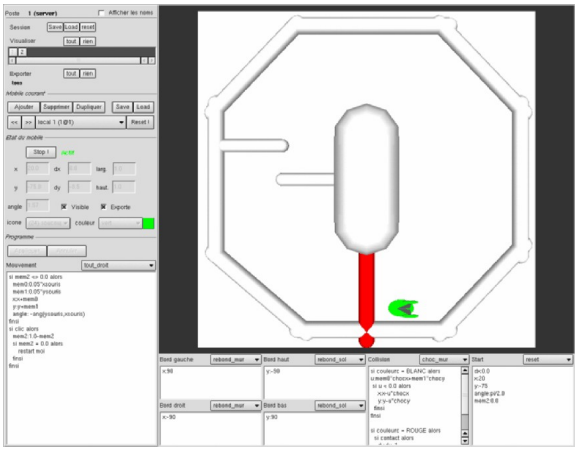


Figure 2: The MobiNet worksheet

MobiNet allows the application of mathematical and physical notions seen in class in a concrete and pleasant way. Through tutorials and exercises, the students discover how a complex process can be decomposed into simpler elements which can then be formalized. Thanks to this approach, students can apply their knowledge concretely and form complex applications, such as the simulation of physical phenomena or video games. We also encourage collaborative work.

1.2. Project History

For several years scientific curriculums have been attracting less and less students. The Senatorial report entitled "*scientific and technical culture for everyone: a national priority*" points out that the decay measured at university has its origin at high school. In this context in 2002, INPG founded a biannual *Engineering Week*. This event consists in inviting high school classes to discover the engineering disciplines and the world of scientific research. During this week, the students participate in half-day activities on various themes. The MobiNet project was developed as a platform for one of these in the iMAGIS team of GRAVIR laboratory. It has been conceived, realized and benched by a team of two faculties and two students (one of which – Sylvain Lefebvre – realized most of the work as part of his teaching assistant project (French *monitorat*)). Since 2002, December, 16 sessions (batches consisting of half a class, during an half-day) have occurred in the *Virtual Reality Classroom* (ARV) at ENSIMAG-Montbonnot.

2. Pedagogic Motivations

The purpose of MobiNet — as presented to students — is to explain how video games are created. Video games are indeed an excellent subject to teach programming, math, and physics, since they involve all the three in their conception. Also, it is interesting to to show to the students how a real world complex object (the video game) can be broken down into smaller parts that they can understand – and even create. Moreover, most video games are based on a simplified simulation of the real world. To build a game, it is therefore necessary to create a model of the world, which is the base of the *scientific approach*. Finally, working on a subject such as video games creates a strong motivation for the students. The students are thus more willing to do exercises using our software. This motivation is increased by the fact that they can create their own network game and play with it.

Behind this face-value purpose, deeper goals also exist:

- Putting theoretical notions learned in class into practice (since formulae are used to create motion of objects).
- Introducing the scientific approach to the student. Visible objects in MobiNet are characterized by different attributes. These attributes can be measured and correspond to *state variables* of the objects. This provides a *mathematical model* of the real world. Moving an object corresponds to modifying its state variables with respect to time. This is expressed with a set of formulae (e.g., `x : x+1` , or `x : 10*cos(t)`) which describes a *physical model* of behavior for the mobiles.

The underlying motivations of our approach:

- The importance of experiencing notions is well known. A concept is usually better learned if it has been put into practice.
- Having a practical goal (creation of a video game) is a strong motivation for the student.
- Playing with the parameters of a formula and having a visual result develops an intuition about the formula. The student better understands the link between the numerical value of a parameter (or its changes in value) and the resulting physical or mathematical behavior (amplitude, frequency, scale, ...).
- Students have to learn by trial-and-error (no bad scores for errors!). We encourage them to understand and explain the unexpected results they obtain.
- The approach proposed by MobiNet corresponds to the approach used by engineers, programmers and researchers in their everyday work. Our purpose is to show to the students how the knowledge they are learning has applications outside of the school world and corresponds to current activity in several professions.

To reach these goals, we chose to not include any physical laws into MobiNet. Every behavior or phenomena to be simulated must be explicitly programmed (gravity, dynamic, even limiting to the screen borders).

3. The MobiNet Platform

3.1. Description

MobiNet is based on the manipulation of graphical objects called *mobiles*. A mobile is described by a set of *state variables* and its behavior (i.e. changes of state variable's values over time) can be settled directly by the user or controlled through a formula. Complex applications can be built by combining some simple mobiles (see Figure 1). Indeed, we provide a complete programming language[1] to describe the evolution of the state variables of the mobiles and the mobiles interactions. Formulae setting the state of a mobile can even depend on those of another mobile (possibly over a network). Figure 2 and Figure 3 detail the user interface and the program areas.

Throughout the software conception we have kept in mind that this is a tool for non computer scientists. Therefore the interface is very simple and the user doesn't need to understand all the possibilities of the software to be able to use it.

[1] Available in French and English.

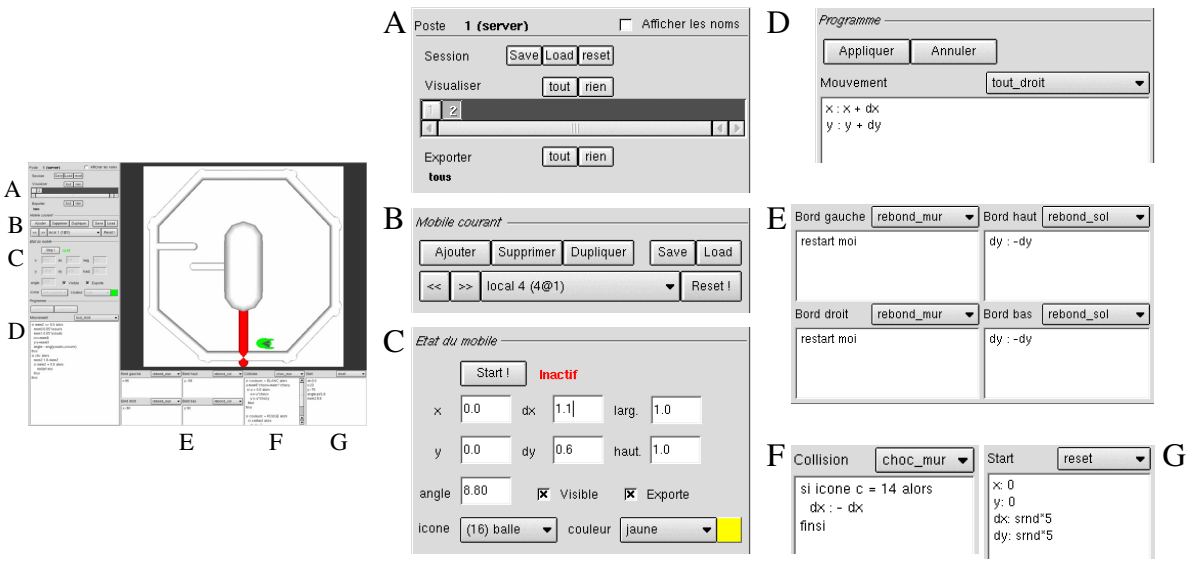


Figure 3: Close up view on the user interface.

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.
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.

The programming language has been defined in collaboration with mathematics and physics teachers in order to suit a high school level. For example we have chosen to use ':' for assigning a value instead of the C-style '='. We kept the syntax as simple as possible and as close as possible to the notations used in class (no '{' or '[' ugly characters !). We have also included a lot of predefined functions in order to accelerate programming when a notion is well understood by a student (e.g., `dist(m1,m2)`).

A key characteristic of MobiNet is that it can export and share mobiles through a network. On a computer the user can see and interact with the mobiles of the other connected machines – if they allow it. Thus it is possible to build exercises based on a collaboration between two or more machines. For example with a pair of computer one can easily program a "pong" game (the simplest tennis game you can imagine). In our experiments, we have found that the collaboration between different computers is an excellent motivation for students.

We also provide special mobiles (which are invisible) to ease user interaction with the mobiles. For instance the special `mouse` mobile always follows the mouse pointer. By reading the `x,y` state value of the mouse mobile, one can easily attach one's mobiles to the mouse. Similarly, camera and lights are special mobiles, in which state variable can be modified (e.g., to follow a simulated mobile). It is also possible to read keyboard inputs and mouse buttons from mobile programs.

Mobiles are 3D textured and lighted objects lying on a 2D plane. Indeed, we imposed this limitation to 2D to ease the interaction (e.g., collisions), especially over a network. However, we could easily manage 3D coordinates if necessary.

3.2. Examples

MobiNet can be used to create a wide variety of applications. Figure 1 depicts some of the examples included with the MobiNet distribution:

- animated solar system,
- simple video games (pong, asteroid, arkanoid, car race...),
- animated geometrical diagrams (parabola, inscribed circle of a triangle..)
- simulation of a physical setup (spring, Galton experiments, marked trajectories...)
- simulation of the propagation of nervous influx,
- simulation of an articulated body with muscles,
- etc...

4. Pedagogic Use

4.1. Experiment: MobiNet tutorial sessions

As explained in Section 1.2, we organized half-day activities around MobiNet in the context of the *Engineering Week*. The duration of a session is less than three hours, and it is designed for between fifteen to twenty students, each (or each pair) having access to one networked computer. The first half-session (1 hour) consists in helping the students discover the interface and the language of MobiNet. For that we mix information and small exercises. Our approach is to let them play with the software and make mistakes. Each time they try something which does not produce the expected result, we encourage them to understand why their mobile behaves thus and to explain how to fix it to get the desired motion. Note that several of these exercises are direct use of what should be known academic knowledge: moving in a straight line, or in circles, placing an object 3 centimeters to the right of the mouse, or moving symmetrically to the mouse...

After this first half-session we take a short break (some won't go !). Before they leave we give them an overview of the second half and ask them to be prepared to form groups of two machines.

The next part of the session is dedicated to the creation of a pong game to be played through the network. Each group creates a pong game which is simulated half on one machine and half on the other. The students can play together with the pong game they create at the end of the session.

Each student of a pair is put in charge of one side of the pong game terrain (left = red player, right = blue player). He inserts a goal on the terrain with appropriate color, orientation and position. He also has to create a racket and a ball which bounce off the walls (of course it is preferable to keep only one ball for the final play, but this is not compulsory !). The detail of the pong game implemented on MobiNet is depicted Figure 4. Note that it can easily be adapted for three players (one horizontal).

The role of the teacher and assistants
We organize the team around one teacher in charge of global explanations and the announcement of new exercises. To help his task, he (or an assistant) manipulates MobiNet on a machine (the MobiNet server) on which the screen is duplicated to a video projector. During the exercises, any student screen (possibly all) can be projected instead (by simply using the import feature on the server). At the beginning of the session, he presents the assistants and their status (ideally, men and women from different curriculum), the MobiNet project and the purpose of the tutorial session. His interventions become less and less frequent as the session progresses in order to let the students experiment personally with MobiNet.

The rest of the team is in charge of helping the students through the various exercises. Usually the team is composed of three assistants for every ten computers. Their first goal is to prevent the students from being slowed down in their progress by a lack of computer skills, the interface or the language. However their most important task is to encourage students (at the beginning some don't dare to type anything), to help them *understand* their errors and to make *the link between* the *formulae* they enter and the *observable behavior* of the mobiles.

Student's reactions
More than 300 high school students participated in MobiNet tutorial sessions between 2002 and 2004. Students came from different high schools (with various ranks) of the Grenoble area, mostly 10^{th} grade (French *seconde*) and occasionally 11^{th} grade (French *première*). They did not have any specific knowledge in Computer Science nor any previous preparation for the session.

Typical behavior can be observed during the session. Some students try to think about the exercise on paper and proceed step by step when implementing their solution in MobiNet. Others just click everywhere, trying every accessible button, until they realize that without some organization they are not going to achieve the goal of the exercise. Among all these behaviors, there are two that underline how a platform like MobiNet can be of help for different kinds of students:

- Some students try to solve exercises by recycling a ready-made solution to a classical exercise learned in class (e.g., finding a line equation) instead of really thinking about the concrete problem in front of them. After an adaptation period, many of these students discover through MobiNet that concepts learned in class can be used as *tools* to solve practical problems.

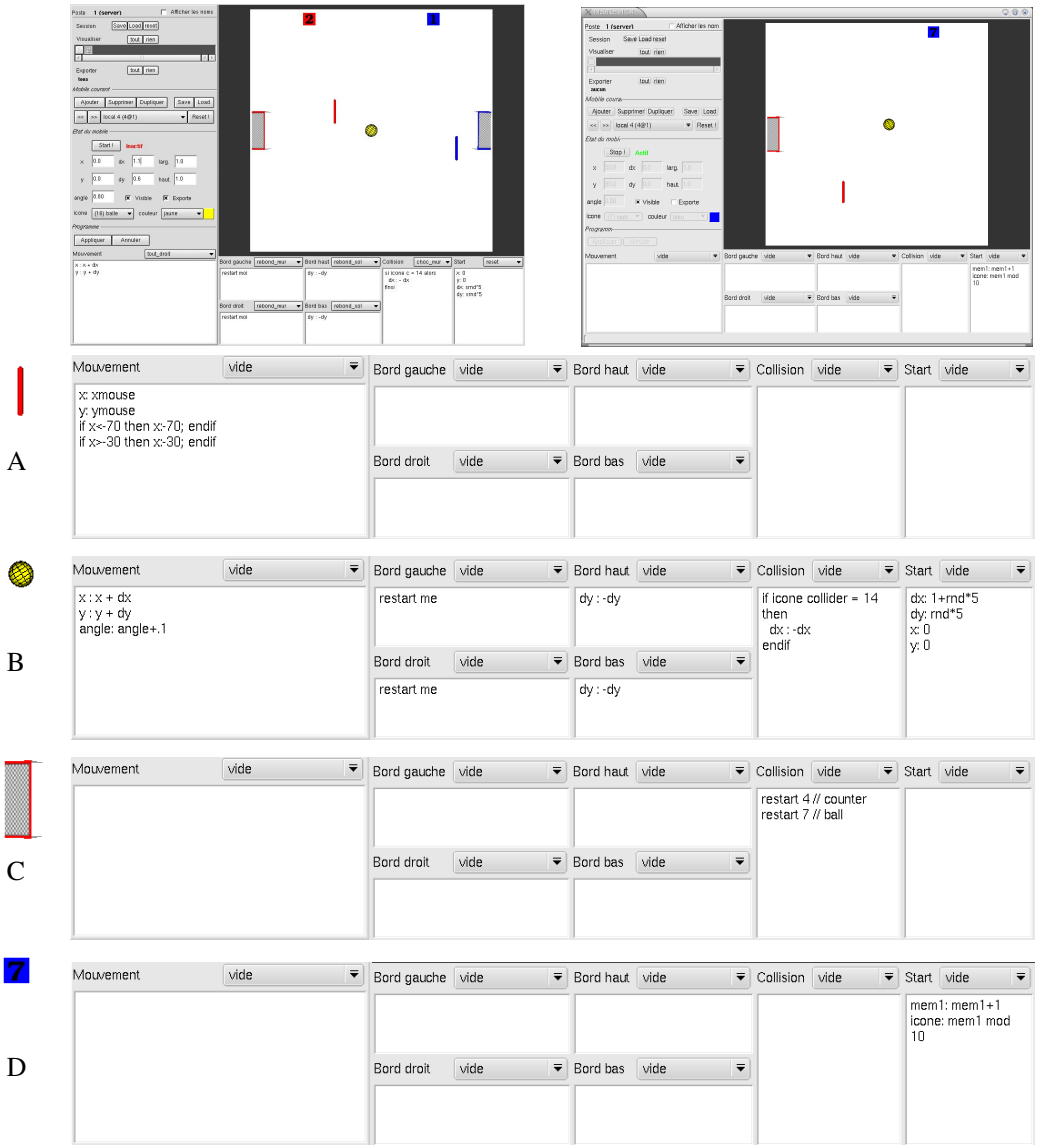


Figure 4: The pong game implemented in MobiNet

Upper Left: Screenshot of the game running. Note that half of the mobiles are simulated by the remote machine.
Upper Right: The mobiles created by the left player (red). Note that the red player creates the score counter for his opponent.
A: The racket mobile. The movement follows the mouse with a positional constraint.
B: The ball mobile. The ball moves in straight line according to the direction defined by the `dx,dy` state values. It also has a rotation motion. When the ball is restarted, `dx,dy` are initialized with random values and the ball is positioned at screen center. If the upper or lower border is reached the `dy` variable is inverted, making the ball bounce. If the left or right border is reached the ball is restarted. If the ball collides with a racket (`icon collider = 14`) then dx is negated so that the ball bounces off the racket.
C: The goal mobile. The goal is positioned by setting its position state values `x,y` manually. If the ball collides with the goal, the counter mobile is restarted (to be incremented, see below) and the ball is restarted. Only the ball can collide with the goal thus no `if` is required.
D: The counter is positioned by setting its position state values `x,y` manually. Its initial shape is set to the "zero" icon. Each time the counter mobile is restarted, its state variable `mem1` is incremented. The icon is set to `mem1 mod 10`. Since the icons representing digits have the same id number as what they represent, the value displayed corresponds to mem1.

The programming language is available both in French and English. Note that (lockable) presets menus contain valid examples for each area.

- Some students are afraid to try and manipulate the software because they think they are not capable of doing math (and often denigrating themselves). Usually they try to hide in a group of two and let the other student do all the work. Once detected and installed on a separate machine (despite protestations!), and after we guide them through the first steps, they usually realize that they *can do* something with their knowledge and start to enjoy playing with programming MobiNet.

We were surprise to see that what should have been well assimilated base knowledge was often not mastered even by "good" students: E.g., which operation should be done to displace a mobile 3 centimeters right of the mouse: add or multiply ? How to modify the amplitude or speed of $sin(t)$? How to get a mirror symmetry ? Which variable should be modified at a bounce: `x` or `dx` ? Throughout the tutorial, students were able to build *intuitions* for themselves from the concepts.

Most of the students were very satisfied with this experience. Some of them even skipped the half-session break to continue exploring the possibilities of the software. However, there are a few students are not motivated and are reluctant to go through the exercises (still too much math). The next section provides more details on the evaluation of MobiNet by the students.

4.2. Evaluation

After the first two MobiNet weeks (in December 2002 and April 2003) we asked the students to fill in an evaluation form. Three classes replied (a total of 77 students): the 10^{th} grade and 11^{th} grade (French *secondes* and *premières*) from *Pablo Neruda* high school and the 10^{th} grade from *Grésivaudan* high school.

Each question had a numerical answer ranging from 1 to 5 (the most positive) and could be followed by a comment. There were a lot of comments. Some did not correspond to the numerical evaluation (!). Figures 5 to 10 depict the results for all the students involved.

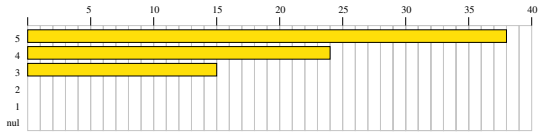


Figure 5: *Did the MobiNet tutorial session interest you ?*

As shown by the results of Figure 5, the students were all interested in MobiNet. The comments underlined the fun aspect of the platform and the interest in understanding how video games are created. Being able to create their own basic games was also greatly appreciated by the students.

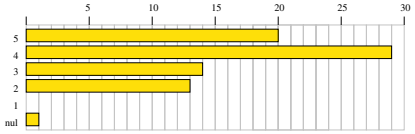


Figure 6: *Would you say that this tutorial session was useful ?*

Overall, most of them felt that MobiNet was useful (Figure 6). Students gave the following reasons: introduction to video game programming (16 comments), initiation to Computer Science (13 comments). A few students were not interested at all (3 comments), which they justify by a lack of interest towards computers in general.

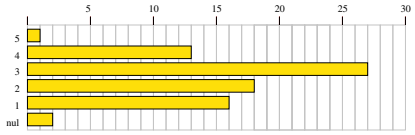


Figure 7: *Before beginning the session, did you think it would be easy ?*

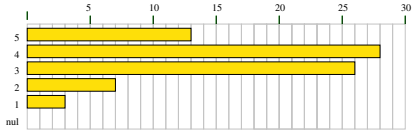


Figure 8: *Did you find MobiNet easy ?*

Most of the students found MobiNet easier to use than they expected (Figure 7 and 8). They especially appreciated the documentation and the interface (19 positive comments against 3 negatives). Some found it difficult at the beginning but they underlined that it became easier after a few exercises (11 comments). A small group of students (9 comments) had some difficulties with the required mathematical background (variables, slope equations, trigonometry). For a few students these difficulties persisted for the whole session.

Overall this results are encouraging. We were expecting more difficulties due to the number of new notions introduced in the first hour. However, it also shows that students have a hazy understanding of many fundamental notions – something that we observed during the session (see the previous section). We interpret this lack of deep understanding to a lack of experimentation. There are too few links between taught concepts and the real world.

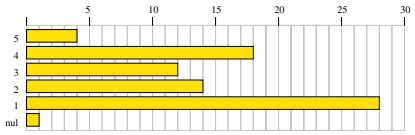


Figure 9: *Did this session change your point of view on math and physics ?*

We asked the students whether this experience changed their point of view on math and physics (Figure 9). Three profiles can be distinguished. First, there are the students who discovered that there are real applications behind the theoretical knowledge taught at school (16 comments). For these students, our goal has been reached. Secondly, there are the students who were already interested in math and physics and did not change their point of view (9 comments) – which is good ! Finally, the students who did not see the link between MobiNet and the math and physics classes, or the students who are discouraged by the mathematical aspect of MobiNet (11 comments). As depicted on Figure 9, most students claim to not have changed their point of view. However one third of them say they have been positively influenced. This tendency is actually stronger for the 10^{th} grade (French *secondes*). This is good news since in France students have to choose whether to pursue their study in science, literature or economics after the 10^{th} grade.

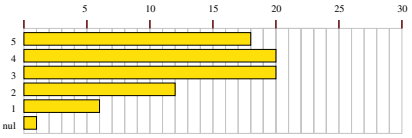


Figure 10: *Do you think using MobiNet in high school could be useful ?*

We finally asked the students whether they thought MobiNet could be used in high schools. Overall answers are encouraging. The students are motivated by the fun aspect of MobiNet (14 comments). They think that MobiNet could be used to help understand notions seen in math classes (11 comments). Students also ask for an access to MobiNet in computer labs, or propose to organize clubs around MobiNet (13 comments). Some think this would be difficult due to the lack of equipment in their high school (11 comments). We discuss further the various possibilities for the future use of MobiNet in section 5.

It appears that most students did not realize that the notions taught in class were directly related to engineering and the description of the real world. This unexpected usefulness motivates "average" students who were not convinced that abstract notions were worth the effort of learning. For "good" students, it strengthened their already existing interest in science. Seeing that it corresponds to real careers might encourage them not to abandon the scientific disciplines. For a few students with difficulties, the mathematical level required by MobiNet was sometimes still too high. Moreover, these students are usually not motivated by working with computers. For those students, other approaches in smaller groups have to be studied. For most students, MobiNet was an entertaining and interesting experience. However many of them did not see the link between what they learned at school and MobiNet: it was too fun to possibly be the same academic subject than taught in class !

5. Other Pedagogic Usages - Future Work

We mostly experimented the use of MobiNet in the context of a tutorial during the *Engineering Week*. But for this, we used just a small amount of the possibilities of the platform:

- At the language level, MobiNet is also able to deal with basic communication between mobiles, mobiles perception (e.g., who is the nearest in a given direction), colored 3D lights, customizable camera, drawing of trajectories, etc. For precise physics simulation, an accurate realtime `dt` value is available.
- At the interface level, it is easy to program mobiles to be used as sliders or buttons, mesh edges or nodes (see examples in the package).
- At the network level, the fact that mobiles of one computer can interact with those of another computer allows for various working modalities: independent task, collective task (in pairs of machines or more, possibly over a long distance), projection of a given screen on a videoprojector (or even of all screens superimposed), the sharing of a common base of simulation prepared by the teacher to be completed locally...

Moreover, here the students were involved out of the context of the class, for only a couple of hours. In our context, we were having a specific series of different goals: showing academic subjects on a different way, giving an initiation to engineering, to the scientific approach, to computer science...

In order to go further and to really bring a meaningful help to the learning of scientific subjects, we really now want to adapt the use of MobiNet *inside* the class – or at least inside the school.

MobiNet can be used in multiple ways in the classroom:

- By the teacher during lecture to visualize a specific notion, or during tutorial to let the students manipulating the parameters in order to get an intuition of how this behaves (e.g., geometric diagrams, parametric curves, springs...).
- In physics lab, a virtual set-up instead of a real one (often costly and limited) can be prepared by the teacher and manipulated by the students. This can probably also apply for other subjects such as biology, technology or math.
- MobiNet can also be used for students projects (e.g., French *TPE*), or possibly homework whenever feasible.

- In particular, 6 junior high-schools around Grenoble are currently testing the *electronic schoolbag* (basically, a laptop allocated to each student and featuring a few dedicated softwares). How to use this opportunity is still a challenging question for teachers. It would be really interesting to see what can be done with MobiNet in this context and how to adapt it to the age of junior students.
- More generally, it would be also interesting to investigate how it could be used with young children (e.g., primary school), or at University out of Computer Science curriculums.
- Of course MobiNet can be used for personal purposes at home or within a student club in the school, e.g. to develop various projects such as simulations or video games.

For structured activities (lectures and tutorials) it is really important to conceive *pedagogic scenarios*: A classical trap with using high tech tools at school is to get students having fun for a couple of hours without having advanced equivalently in the official educational program: In such a case the technological activity is an *extra activity*, and not really a part of the course. It is especially challenging to select adapted units of knowledge and to build pedagogic scenarios illustrating them.

We would like to assemble a repository of activity modules freely available to teachers. This task can be done only by teachers, or in tight dialogue with teachers. So we would like to establish working groups, and a network of users within teachers (through dedicated mailing-lists and repositories). The fact that we kept contact with the high schools participating to the *Engineering Week* will help us establishing a teacher network. A collaboration should start soon with a team at IREM (*Research Institute of Mathematics Teaching*), and two new students will do their teaching assistant project (French *monitorat*) with us.

Acknowledgment

We thank INPG for the happening of the *Engineering Week* event which provided us with the occasion to create the MobiNet project, and particularly Maryse Beguin for her constant support. Thanks are also due to the numerous students and colleagues who accepted to help us as tutorial assistants, and particularly Patrick Kocelniak who is in charge of the *Virtual Reality Classroom*. Of course, we want to thanks the high school students and teachers who participated to the operation. Thanks are also due to David Roberts and Elmar Eisemann for rereading the paper.

Despite the project was out of our official research topic (thus requiring time over the already busy schedules!), the universal enthusiasm we encountered both from users and academic responsibles (at IREM, CCSTI, Greco, GTD-math, INPG...) greatly contributed to the dynamics of this project. We hope it is just the beginning of a long story that we will be happy to share with other volunteers.

Distribution and Contact

The MobiNet platform is free software under GPL license. It is available for Windows and Linux at `http://www-imagis.imag.fr/mobinet/`.
It is provided with a manual and a full set of examples. Note that the interface description, language description and list of examples are also available on the web site, together with various extra documents. The programming language is available in French and English.
The team can be contacted at `mobinet@imag.fr`.
Moreover, we propose several mailing-list (for announcements, for teachers, for developers).

