MobiNet - The interface (v1.1)

Networked platform for mobiles programming.

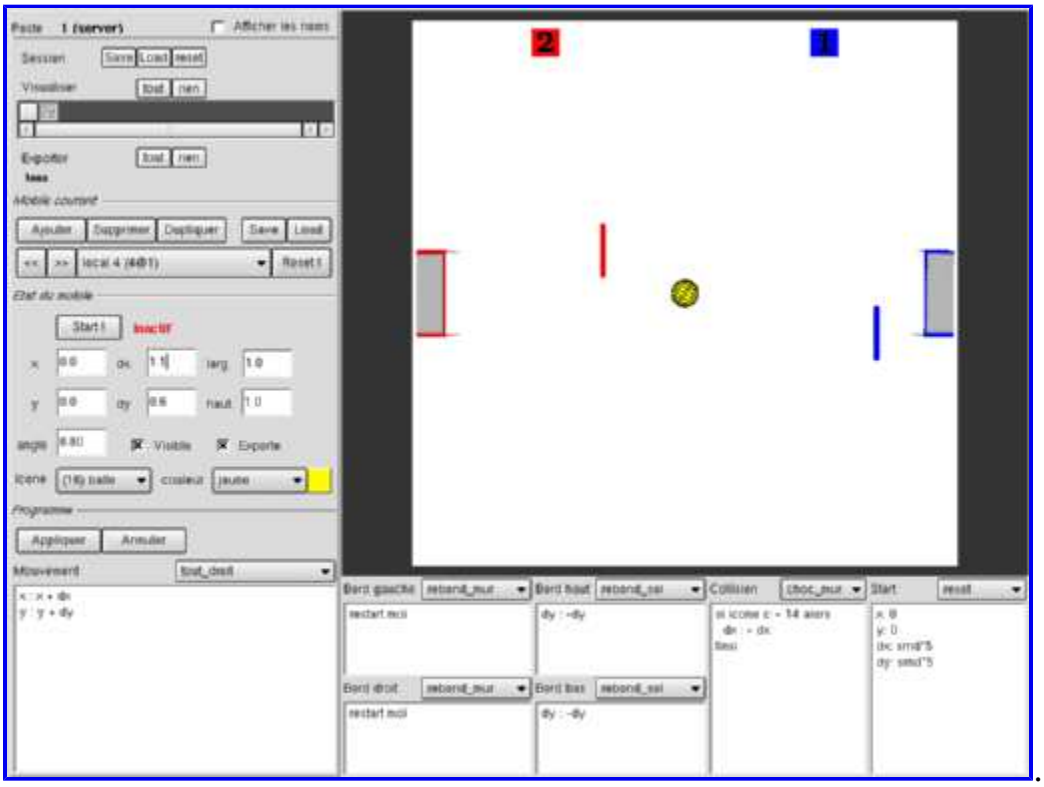
Back to MobiNet home page.

*In MobiNet, one programs the behavior of various **mobiles**: These are the various visible objects (whether they move or not!). The interface allow the tuning of every behaviors of the current mobile: aspect, motion, colliding rules with borders or other mobiles... (then one switch for a mobile to the other).*

General aspect of the screen (click for zooming). The various areas are described below.

Here, we programmed 7 mobiles: the ball, the racket, the goals, the counters.

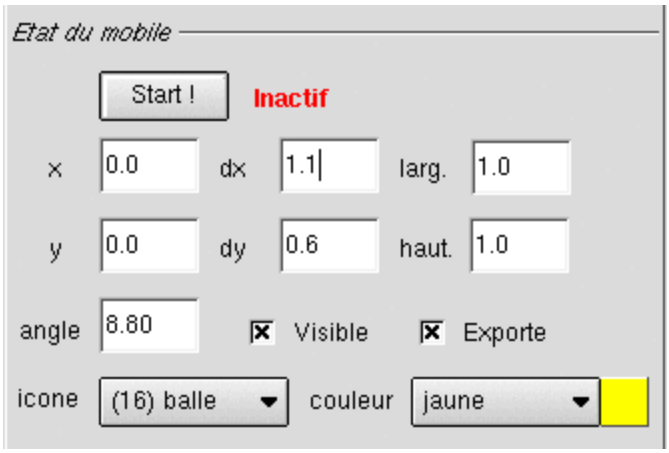
- The ball follows a straight line trajectory, and bounce on borders and on the rackets.
*- The rackets follow the mouse vertically. In facts, one of the rackets (and the corresponding goal and counter) is managed by **another computer**: here, two users share their mobiles on the network. One manage the reds and the ball, the other manage the blues.*
- The goals account for collisions with the ball, and trigger the opposite counter.
- The counters increment when triggered by the goals.



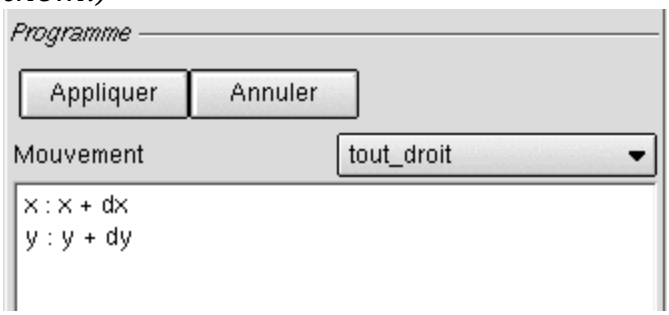
By clicking on `display names', one can see the mobiles number (`local n` *when managed on the computer,* `n@m` *when managed by computer m*) and the

coordinate system (*-100 to 100 for x and y*).

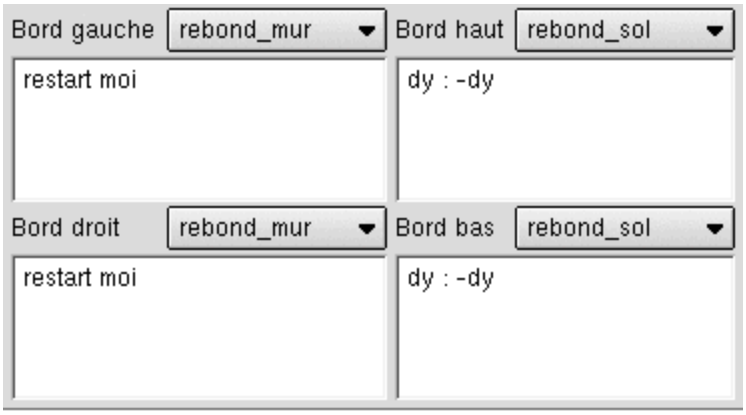
- Parameters describing the current mobile (*here, the ball*): position, dimension, icon, color, orientation... (*moreover, we will use dx,dy to encode the motion direction.*)
NB: in fact all is number (*e.g. the ball icon is number 16*), which allows calculation.



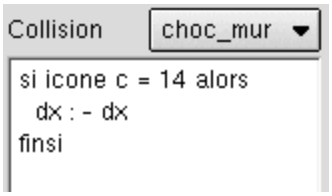
- Programming the motion: it consists in describing the changes to be done between 2 images (about every 25th second) by "`such_parameter: new_value`". See the language manual for the set of usable **commands** in the program areas.
One can set small iterated steps (`dynamics'), describe a function of time (`cinematics'), or of the mouse, or of the position of the other mobiles (e.g. chasing), or any combination programmed according to once fantasy.
*(NB: predefined sets of behaviors are also available to pick up ingredients if wanted. These **presets** are stored in a text file, so an teacher can easily modify them.)*



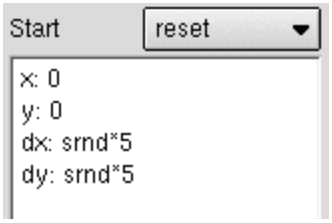
- What to do when a mobile touch a border.



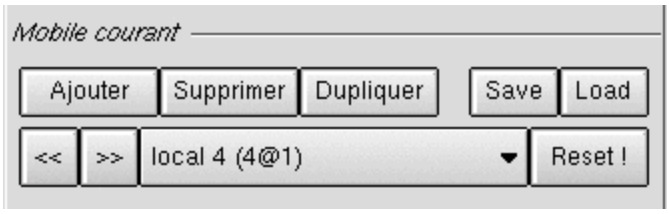
- What to do when a mobile touch another mobile.
If necessary, one can settle a different behavior, depending on the collider identity.
(e.g.: *if collider's icon is a racket then...*)



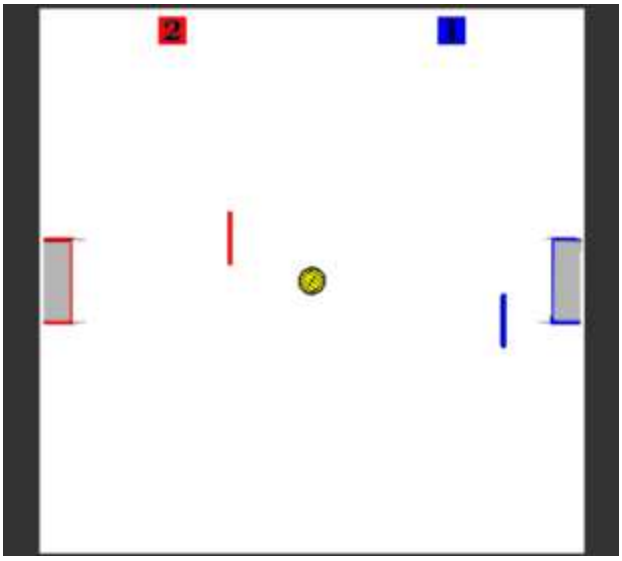
- Parameters value when (re)starting.
NB: a mobile can ask to restart itself (e.g. in case of collision) or to restart another mobile (e.g. to trigger a counter, a shot, to kick-off the ball...) using the command `restart num`. It is also a way to send **messages** between mobiles.
(`srnd` *gives a random number between -1 and 1.*)



- Creating a new mobile, choosing the current one, etc.

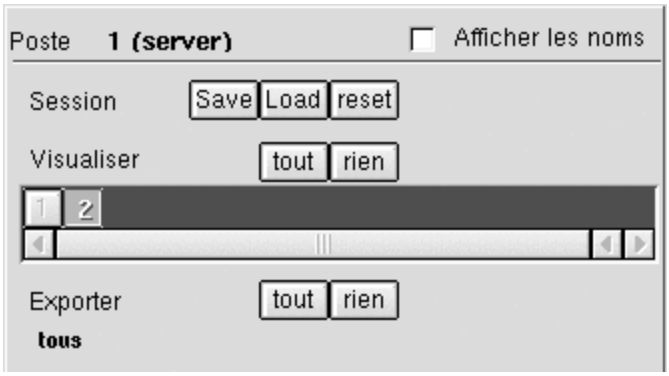


- Resulting display.



- Reading, writing, interacting with other computers through the network... One chooses the mobiles to be **exported** (i.e. that one make visible on the network) and the computers whose we **import** the mobiles.

Thus, several working modalities can be used:
- independent work on each station.
- displaying any station from a master station.
- displaying every station superimposed (e.g. pour wall projection).
- collaborative work by groups if 2 or 3 stations (or more).
- collaborative work between N stations (not necessarily on the local network).



- That's it ! Quite easy, isn't it ?
