Sonigraphical Instruments: From FMOL to the *reacTable**

Sergi Jordà
Music Technology Group
Pompeu Fabra University
Ocata 1, 08003 Barcelona, Spain
+34 93 542 2104
sergi.jorda@iua.upf.es

ABSTRACT
This paper first introduces two previous software-based music instruments designed by the author, and analyses the crucial importance of the visual feedback introduced by their interfaces. A quick taxonomy and analysis of the visual components in current trends of interactive music software is then proposed, before introducing the *reacTable**, a new project that is currently under development. The *reacTable** is a collaborative music instrument, aimed both at novices and advanced musicians, which employs computer vision and tangible interfaces technologies, and pushes further the visual feedback interface ideas and techniques aforementioned.

Keywords
Interactive music instruments, audio visualization, visual interfaces, visual feedback, tangible interfaces, computer vision, augmented reality, music instruments for novices, collaborative music.

1. INTRODUCTION
For the last ten years, my main area of interest and research has focused around the possibilities for bringing new musical creative facilities to non-musicians, without degrading neither the music potentially producible, nor the users' interactive experiences and control possibilities. Moreover, and because of my penchant for free-jazz and improvisation, I have chosen to concentrate on real-time interactive solutions, which I also feel can be more suitable (i.e. more easily encouraging, exciting and rewarding) for the non musicians than the more thought demanding non-real-time compositional tools.

New musical tools or instruments designed for trained musicians, or even for specific performers, can be quite complex and challenging; as a counterpart they may offer a great amount of creative freedom and control possibilities to their players. On the other hand, instruments designed for amateur musicians or for public audiences in interactive sound installations, tend to be quite simple, trying in the best case, to bring the illusion of control and interaction to their users, while still producing "satisfactory" outputs. Logically, these two classes of instruments are often mutually exclusive. Musicians become easily bored with "popular" tools, while casual users get lost with sophisticated ones. But is this trend compulsory? Wouldn't it be possible to design instruments that can appeal to both sectors: tools that like many traditional acoustical instruments, can offer a *low entry fee with no ceiling on virtuosity*? [53] With these questions in mind I started in 1997 the conception and development of FMOL, a path that has recently taken us to the *reacTable**.

2. SONIGRAPHICAL PRELIMINARIES

2.1 Epizoo (1994-1995)
Several years before FMOL, together with the visual artist and performer Marcel.lí Antúnez I had developed the computer-based interactive performance *Epizoo* (1994-1995).

The project was not a *musical instrument*; at least not only. Integrating elements of body art, videogames and multimedia applications, it allowed volunteers from the audience to play with (or "tele-torture") the performer's (i.e. Antúnez's) naked body via a graphical interface [29][34][45]. *Epizoo*'s graphical interfaces, as seen in figure 1, could seem to come from a weird videogame designed by the likes of Hieronymus Bosch or Giuseppe Archimboldo, but the fact is, that these GUIs still stick to the typical, hypertextual multimedia cd-rom or web approach: buttons (albeit very hidden) for discrete selections, and sliders (or hot-spots that evaluate mouse activity) for continuous controllers.

Epizoo musical output was mostly based on wavefile loops and MIDI sequences; loops were often layerable and pitch-changeable, and sequences could be sometimes manipulated in several ways, but each of *Epizoo*'s screens (there are about 15 screens in the complete performance) can be considered in fact more as a musical piece or composition, which happens to have different performances every show, than a true musical instrument. Besides, volunteers did really conduct all the show development, including the music and the lightshow, and they did so through its quite peculiar mouse-driven GUI, but the opportunity to manipulate a real human body seemed to mask all other "banal" interaction possibilities. This, combined with the fact that these users (which could typically have many different "mouse-skills") were being exposed to the interface for the first time, but were still responsible of conducting a show in a cathartic atmosphere, closer to a rock concert or a techno rave than to a *typical* interactive installation, turned *Epizoo* (at least its musical part) into a perfect example for the category earlier exposed: interactive sound installations which promote the user's illusion of control while guarantying their musical output. Whatever the user did, s/he could feel the control over the whole show, but at the same time the output, especially the musical one, would never be "too bad". FMOL was not going to be about that.

Figure 1. In EAX, one of Epizoo's screens, the eyes follow the mouse. The eyes, mouth, ears and legs are hot spots that can be touched and clicked

2.2 Reintroducing FMOL (1997-2002)

FMOL, a project I started in 1997 when the Catalan theatre group La Fura dels Baus proposed to me the conception and development of an Internet-based music composition system that could allow cybercomposers to participate in the creation of the music for La Fura's next show, supersedes most of *Epizoo*'s musical limitations. The FMOL project has evolved since its debut, and several articles have been written that should not be repeated here. The fact is that FMOL exemplifies several paradigms which can be treated independently. It is primarily a tool for collaborative musical composition on the Internet. This feature that was the motto of the initial project is better exposed in [37], which deals with the social and aesthetic implications of net-music, and [38] which cover more technical aspects of the implementation. Furthermore, implications of computer and web based collective or collaborative music composition and performance, starting with *the League of Automatic Composers* in the late 70s [9] have been widely studied and published in these last years in papers and thesis such as [8] and [27].

Technical aspects of the FMOL software (real-time synthesis engine, etc.) are covered in [35]. The didactical, intuitive and proselytizing aspects of FMOL as a tool for introducing newcomers into experimental electronic music are deeply treated in [41], while [39] or [40] also cover its use as a professional instrument and its attempt at dealing simultaneously with *micro-sonic* and *macro-musical* compositional ideas.

In this paper I want to focus only on the peculiar aspects brought by FMOL's unique user interface, which presents a closed feedback loop between the sound and the graphics: in FMOL, the same GUI works both as the input for sound control and as an output that intuitively displays all the sound and music activity. After explaining deeper this idea, I will discuss different ways where these sonic-graphic relations are present in recent audiovisual software, and the path that has led us to the conception of our new project, the *reacTable**.

2.3 FMOL Musical Output

With FMOL I wanted to introduce newcomers to experimental electronic music making. Therefore, for obvious availability reasons, the instrument had to be a mouse-driven software (it can still be freely downloaded at [22]). I also wanted to create a simple and complex tool all at once; a tool that would not dishearten hobbyist musicians, but would still be able to produce completely diverse music, allowing a rich and intricate control and offering various stages of training and different learning curves.

Both goals have been, in my opinion, quite well attained. During the two Internet calls for musical contributions for two of La Fura's shows (January-April 1998 for F@ust 3.0, and September-October 2000 for the opera DQ) more than 1,700 compositions were received in the database [23]. We know now that many of the participants had no prior contact with experimental electronic music and that a few were even composing or playing for the first time, but the final quality of the contributions (which can be heard online, as well as on the the Fura dels Baus' F@ust 3.0-FMOLCD published in 1998 [37], and on the more recent CMJ 2002 companion CD [51]) was quite impressive.

Moreover, I have given several FMOL workshops usually with a mix of musicians and non-musicians, and if the feeling is positive they usually end with public concerts. An improvisation fragment recorded after one of these workshops can also be heard in [51]. The intuitiveness acid test took place in March 2003 during a one-day workshop with 5 to 8-year old kids from Galicia (Spain), which ended with surprising collective improvisations.

It takes about half-hour to start having fun with the instrument, and several hours to acquire some confidence and produce controllable results. However, after five years of playing it, I am still learning it and do often discover hidden features. Because, and that is another important point, it happens that the instrument I originally designed as a cheap and freely available system for "experimental electronic music proselytism", turned to be, to my own surprise, my favorite instrument for live concerts. Since 1999, the FMOL Trio (Cristina Casanova and me on FMOL computers, plus Pelayo F. Arrizabalaga on saxophones/bass clarinet and turntables) performs free-form improvised electronic music and has produced several live CDs [21][24][25][26].

2.4 FMOL Visual Feedback

Arguably, visual feedback is not very important for playing traditional instruments, as the list of first rank blind musicians and instrumentalists (e.g. Ray Charles, Roland Kirk, Tete Montoliu, Joaquín Rodrigo, Stevie Wonder …) may suggest. But traditional instruments usually bring other kinds of feedback, like haptic feedback [12] [30], which is not so often present in digital instruments, at least in the "cheap" ones. Besides, why should not digital instruments use at their advantage anything that could broaden the communication channel with its player? I am convinced that in the case of FMOL, its unique visual feedback has been a fundamental component for its success as a powerful and at the same time intuitive and enjoyable instrument.

FMOL mouse-controlled GUI is so tightly related to the synthesis engine architecture, that almost every feature of the synthesizer is reflected in a symbolic, dynamic and non-technical way in the interface. In its rest position the screen looks like a simple 6x6 grid or lattice. Each of the six vertical lines is associated with one voice generator (FMOL's sound engine supports six real-time synthesized stereo audio tracks or channels), while the horizontal lines are associated with the effects processors (filters, reverbs, delays, resonators, frequency, amplitude or ring modulators, etc.), embedded in each track. All of these lines work both as input devices (controllers) that can be picked and dragged with the mouse, and as output devices that give dynamic visual and "sonic" feedback.

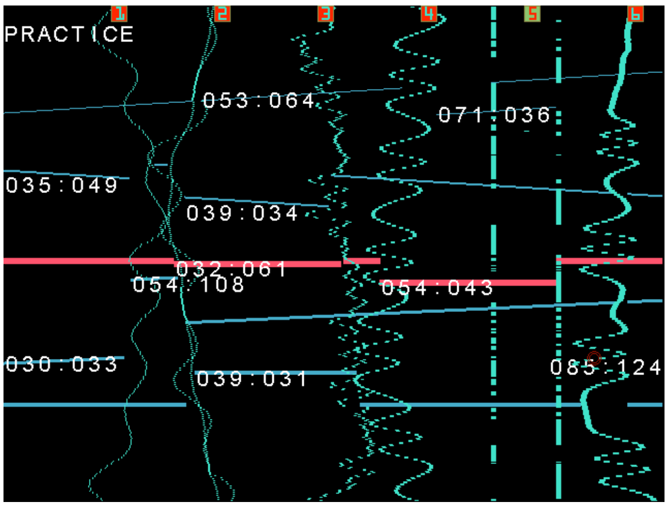

Figure 2. FMOL in action

Mappings and detailed control mechanisms are explained better in [41]. The key point is that when multiple oscillators or segments are active (FMOL engine includes 24 LFOs and 96 parameters to control), the resulting geometric "dance", combined with the six-channel oscilloscope information given by the strings, tightly reflects the temporal activity and intensity of the piece and gives multidimensional cues to the player. Looking at a screen like figure 2 (which is taken from a quite dense FMOL fragment), the player can intuitively feel the loudness, frequency and timbrical content of every channel, the amount of different applied effects, and the activity of each of the 24 LFOs. Besides, no indirection is needed to modify any of these parameters, as anything in the screen behaves simultaneously as an output and as an input.

3. SONIGRAPHICAL TOOLS

3.1 Media players and VJ Tools

In order to show the secular catacomb stage of visual music, Bernard Klein affirmed in his 1927 book *Color-Music: the Art of Light* that "it is an odd fact that almost everyone who develops a color-organ is under the misapprehension that he or she, is the first mortal to attempt to do so" [42]. This assessment could surely not be pronounced anymore nowadays. While since its beginning, digital technologies have boosted multimodality and any kind of parallelism between image and sound in any of their two directions, the truth is that in the last few years, we have seen the flourishing of many software programs or environments that deal with this duality in several ways, even creating distinct families of tools each with its well defined idiosyncrasy.

Following the trend started with the popular music visualization freeware program *Cthugha* released around 1994 and described on its birth as *an oscilloscope on acid* [19], current software music players, like *WinAmp* or *MS Media Player*, come with dozens of fancy visualization plug-ins, that allow the user to view the results of the music in many different ways. These systems can be generally described with the following scheme.

Figure 3. Elements of a standard music player with visualization

However, these systems are not very interactive, except that users can decide to change the visualization plug-in, applying thus a discrete change to block D. When an audiovisualizer of this kind becomes interactive, we have what we could call a VJ Tool [6] [52]. Such tools exist as stand-alone programs such as Jaromil's *FreeJ* [28], *Arkaos* [4] or *Resolume* [48], or can be easily built using visual programming environments like MAX + (Nato or Jitter), or PD + (GEM or Framestein), to name a few of the more popular software combinations.

In this new case, depending on the particular system design, the user could interact at any step of the chain.

Using the aforementioned programming environments, one can also decide to take the complementary approach, and build a musical instrument or a sound synthesizer which can be directly controlled by the analysis of some dynamic visuals. These image input can be of any kind (synthetic, abstract, video, etc.), and can come from any source (stored movies, real-time generated animations, live video input, etc.) Although different in concept, this alternative scheme could also include computer vision based musical controllers.

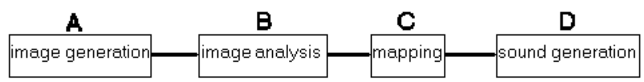

Figure 4. Elements of an image controlled music generator

However, none of these two approaches (Sound→Visuals or Visuals→Sound) does generally close the control loop as FMOL does (i.e. in VJ tools, the way the user modifies the graphics does not affect the music). Besides, they usually present two windows at least: one for the visual output (or input, depending on the chosen approach) and an additional one (the "control panel") for parameter modification; they do not allow to modify the *visuals window* by playing directly on it.

3.2 Golan Levin's Work

To my knowledge, only Golan Levin's work follows an audiovisual approach comparable to the one I've presented in FMOL. The fact is that although we unfortunately did not know about each other until quite recently, I believe our goals and approaches share many common aspects. In his master thesis "Painterly Interfaces for Audiovisual Performance" he proposes a system for the creation and performance of dynamic imaginery and sound, simultaneously, in real-time, and with basic principles of operation easy to deduce, while at the same time capable of sophisticated expressions and indefinitely masterable [43].

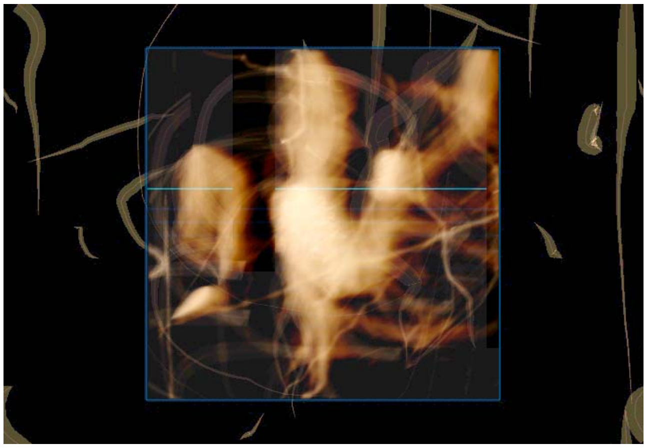

Figure 5. A screenshot of Golan Levin's Yellowtail

Levin talks about an inexhaustible, extremely variable, dynamic, audiovisual substance that can be freely painted, and he has developed many audiovisual tools, like *Yellotail*, *Loom*, *Warbo*, *Aurora* and *Floo*, which follow this path. Perhaps, the major difference in our approaches may be the fact that Levin, like Oskar Fischinger, the animator that in the 40s invented the *Luminograph* color organ [44], is willing to play light while playing sound. For him, image is therefore an end in itself. While for me, it is only the means to an end: a more intuitive interface for creating music.

4. THE *REACTABLE**

4.1 Preliminary

Last year, together with the doctorate students Alvaro Barbosa, Gunter Geiger, Rubén Hinojosa, Martin Kaltenbrunner and José Lozano, and the undergraduate students Carlos Manias and Xavier Rubio, we constituted the Interactive Systems Team, inside the Music Technology Group led by Xavier Serra at the Pompeu Fabra University of Barcelona. One of the initial projects was to port FMOL to Linux and make it open-source, which seemed also a good opportunity for revamping the system [1].

Looking at the way people have used FMOL, and using it myself for improvisation in different contexts and with different musicians, has raised ideas new features and modifications. But we also felt that this control complexity could not be permanently increased; there are limits to what can be efficiently achieved in real-time by means of a mouse and a computer keyboard. Building an external FMOL controller for a faster and more precise multiparametric control seemed therefore a tempting idea. Designing a video detection or ultrasound system that would allow musicians to interact on a big projection screen, grabbing and moving strings with their hands, was the first idea we had. This could surely add a lot of visual impact to live concerts, although we also felt that musical control and performance may not necessarily improve with it. These and other considerations took us to a completely new path, which should profit the knowledge gained during this years and bring it to a much more ambitious project: The *reacTable**.

4.2 Intentions

We aim at the creation of a state-of-the-art interactive music instrument, which should be collaborative (off and on-line), intuitive (zero manual, zero instructions), sonically challenging and interesting, learnable, suitable for complete novices (in installations), suitable for advanced electronic musicians (in concerts) and totally controllable (no random, no hidden presets…). The *reacTable** should use no mouse, no keyboard, no cables, no wearables. It should allow a flexible number of users, and these should be able to enter or leave the instrument-installation without previous announcements. The technology involved should be, in one word, completely transparent.

4.3 Computer Vision and Tangible Objects

As the Tangible Media Group directed by Professor Hiroshi Ishii at the MIT Media Lab states, "People have developed sophisticated skills for sensing and manipulating our physical environments. However, most of these skills are not employed by traditional GUI…. The goal is to change the *painted bits* of GUIs to *tangible bits*, taking advantage of the richness of multimodal human senses and skills developed through our lifetime of interaction with the physical world." [32][50]. Several tangible systems have been constructed based on this philosophy. Some for musical applications, like *SmallFish* [49], the *Jam-O-Drum* [10] [11] [33], the *Musical Trinkets* [46], *Augmented Groove* [7] [47] or the *Audiopad* [5], but we believe that no one attempts the level of integration, power and flexibility we propose.

reacTable* = FMOL + MAX + JamODrum

Substitute if you want *MAX* with *PD* or *JMax* or even *AudioMulch*. Substitute the *Jam-O-Drum* with the table version of *Small Fish*. You can even substitute FMOL, but only with Levin's systems , and you will get an initial idea about what the *reacTable** is all about: a table-based collaborative music instrument that uses computer vision and tangible user interfaces technologies, within a MAX-like architecture and scheduler, and with FMOL-inspired HCI models and visual feedback.

The *reacTable** is a musical instrument based on a round table, which has no sensors, no cables, no graphics or drawings. A video camera permanently analyses the surface of the table, while a projector draws a dynamic and interactive interface on it.

Many interesting and promising computer vision tools, mostly based on body motion capture, are being developed for musical applications [15] [16]. However, many of us do not feel too comfortable "dancing" in front o a video camera (some even without camera!), while we all work and socialize around tables. For this reason our computer vision system does not attempt to track body motion. Instead, it focuses on tracking the hand movements over the table, and on detecting the nature, position and orientation of the objects that are distributed on its surface.

These objects are mostly passive and made out of plastic or wood of different shapes. Users interact with them by moving them, changing their orientation on the table plane or changing their faces (in the case of volumetric objects). More complex objects include (but are not limited to) flexible plastic tubes for continuous multiparametric control, little wooden dummy 1-octave keyboards, combs (for comb-filters), or other everyday objects. In case an object needs sensors, its communication with the host computer will be wireless.

4.4 Visuals (1)

The projection follows the objects on the table wrapping them with auras or drawing figures on top of them. The projection covers also the whole table surface with dynamic and abstract elements that reflect all the system's activity, and depend on the hands' movements and trajectories, the objects' types and positions, and the relations between them all. The projection never shows buttons, sliders or widgets of any kind.

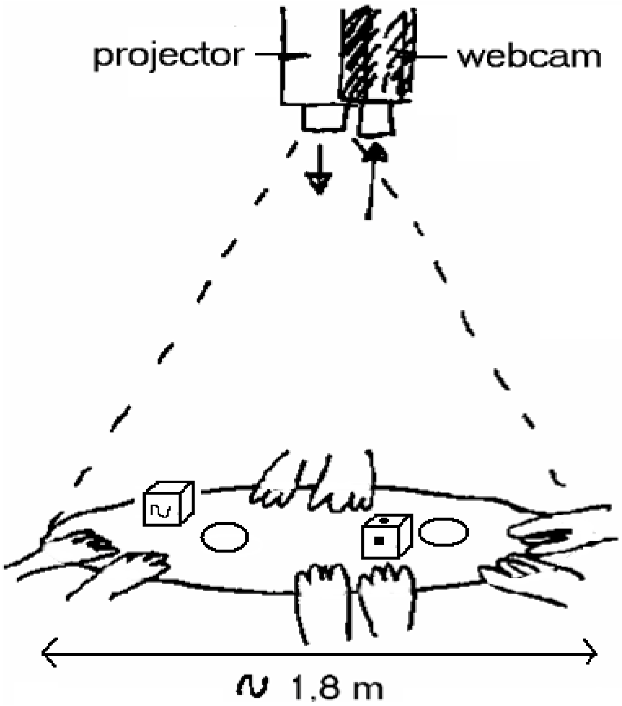

Figure 6. The *reacTable simplified scheme**

4.5 But where is MAX?

FMOL has proven to be quite flexible. Its palette of sound generators and processors includes more than 20 algorithms that (with internal configuration variations) constitute a bank of 127 presets the user can select and apply to any of the strings. This process of "building an orchestra" is not done in real-time while playing, but in a different, more conventional window. Besides, all FMOL macro-control of form is done like in traditional analog synthesizers, by means of LFOs and arpegiators. More sophisticated control sources, such as algorithmic generators, pitch filters, etc. cannot fit coherently into the FMOL interface. The *reacTable** overcomes these restrictions by adapting one of the more powerful real-time computer music software paradigms implemented in the last decades.

Like MAX and all of its cousins, the *reacTable** distinguishes between control and sound objects, and between control and sound connections. Unlike MAX, and more like Audiomulch (which however has no explicit control flux), the *reacTable** objects are more high-leveled; the *reacTable** is an ambitious project but it is an instrument, not a programming language!

When a control flow is established between two objects, a thick straight line is drawn between them, showing by means of dynamic animations, the flux direction, its rate and its intensity. Visual feedback will also guarantee that LFOs and other macrotemporal values will be perceived as blinking animations projected on top of the related objects, showing frequency and shape (e.g. square vs. sinusoidal).

4.6 Visuals (2): Audio flow

Where control flow lines are straight and simple, audio flow lines are *organic* and complex. Their dynamic shapes will show the macrotemporal audio variations (vibratos, tremolos, tempo and rhythms…) and their interior (colors, intensities…) will depend on their spectral audio content.

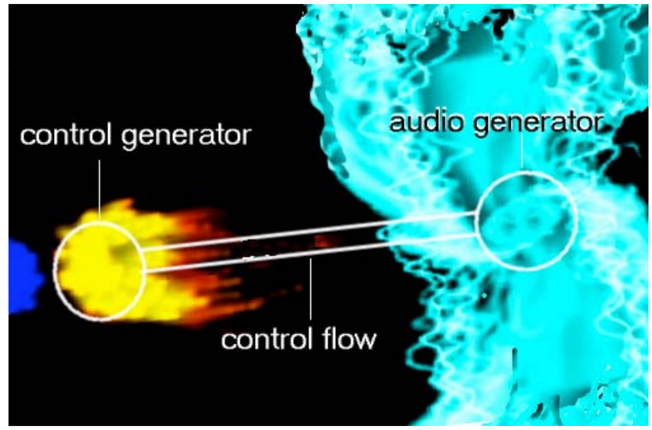

Figure 7. Control and audio flow simulation

Users will also be able to control, modify or fork audio flows without using additional objects, but just by waving their hands, as if they were digging water channels in the beach sand.

4.7 Avoid user's frustration at any cost

To avoid frustrations, a system does not necessarily have to be completely understandable, but it has to be coherent and responsible. Unlike MAX, the *reacTable** has to work "by default" and any gesture has to produce audible results. Here are some of its laws:

- There is not anything like an *editing mode* and *running mode* (at least for *installation users*); the *reacTable** is always running and always being edited!
- Objects are not active until they are touched
- Active objects have a dynamic visual *aura*
- Objects are interconnected by proximity
- If on start-up, a user activates an object that does not sound (i.e. a control object) the closest audio object is automatically linked to it (and the link is visualized)
- Moving an object on the table can change the relations with the other objects
- Relations can also be "fixed" touching two objects with the two hands. Fixed links are shown with a thicker line or a different color.

Perry Cook, in an informal music controllers design decalogue, ironically points that "smart instruments are often not smart." [18]. Although we basically agree with him, we have come to the conclusion that a system like the *reacTable** must show some kind of intelligent behavior. For example, as most of the control objects are adimensional (some, like the dummy keyboard, are not), when one adimensional control flux is sent to an object that can accept different inputs, the system chooses what the best parameters to control in every case are. In another demonstration of intelligent behavior, the system may suggest interesting candidates for a given configuration, by highlighting the appropriate objects (in a manner not to be confused with LFOs).

The *reacTable** wants to be *user-proof*. For instance, it seems natural that after some minutes, people will start stressing the system in different ways, like placing personal objects onto the table. Although it is no possible to anticipate all objects that people may use, some of the more common could be detected (cigarette packets, mobile phones, keys, pens…) and a "funny" functionality could be added to them (e.g. mobiles could generate pitch in a "mobile-fashion").

5. CURRENT IMPLEMENTATION

The *reacTable** project has started in December 2002 coinciding with the foundation of the Interactivity Team within the Music Technology Group (MTG). We are currently working and researching all the main threads in parallel (computer vision and objects recognition, sound engine architecture, interactivity logic, sound visualization, etc.) while designing the core and the integration of all these branches. Computer vision and objects recognition is being carried using both *Eyesweb* [13][14][20] and the Intel ® Image Processing (IPL) and Open Computer Vision (OpenCV) libraries [31]. We will not describe here any of these issues, as we soon plan to devote a whole paper to them. The synthesis engine is being implemented using the CLAM libraries, the open-source, multiplatform C++ libraries for real-time audio being developed at the MTG [2][3][17].

In parallel with these two main productions threads, we are working with a *reacTable** software-only simulator (that runs on both Linux and Windows), which is an essential workbench for defining and refining all of the system laws, evaluating user interaction and objects' connectivity rules, as well as determining the panoply of sound and music objects, their roles, behaviors, the way they synchronize between them, etc. The simulator GUI has been implemented in Java by Martin Kaltenbrunner, while Gunter Geiger is working on its sound engine using PD for quick prototyping. Both modules communicate via TCP/IP, a flexible architecture which also permits multi-user simulation by running different instances of the GUI in different computers.

Figure 8 shows a *reacTable** simulator screenshot, with only four kinds of sound objects: High Frequency Oscillators (circles), Low Frequency Oscillators (triangles), filters (smoothed squares) and time-based effects (squares). Visual feedback is yet very simple; all connection lines are straight and do not suggest therefore any of the information they transmit, except that to distinguish between the two types of connections, audio lines are drawn in dark, while control lines are light grey. Each dark line flowing into the audio sink represented by the black central circle corresponds therefore to an independent audio thread.

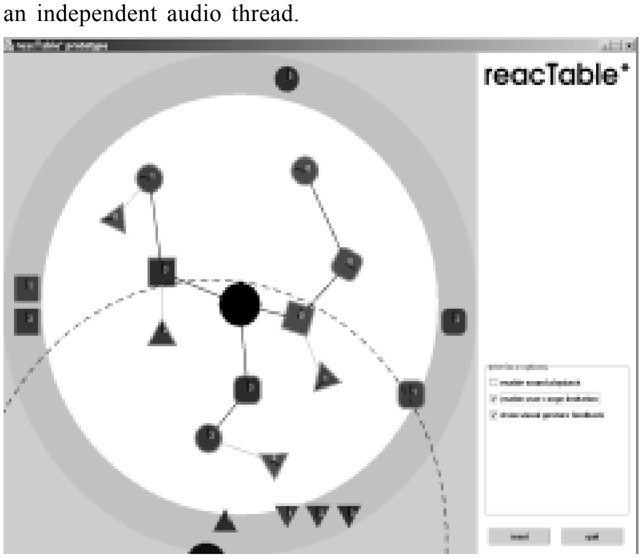

Figure 8. *reacTable simulator snapshot**

At this early stage, what we have is a sort of higher level MAX in which users can drag objects that dynamically interconnect between them according to the rules defined in the system. Using the right mouse button objects can also spin around and connection lines can be broken. Parameters are calculated from the rotation angle of the objects as well as from the length and the orientation of their connections. As simple as it still is, this flexible and dynamic architecture already permits for some fast sound changes that seem impossible to attain in an analog modular synthesizer, which it somehow evokes.

6. FUTURE WORK AND CONCLUSION

The *reacTable** is an ambitious project. Unlike many new designed instruments, its origin does not come from approaching its creation by exploring the possibilities of a specific technology, nor from the perspective of mimicking a known instrumental model. The *reacTable** comes from our experience designing instruments, making music with them, and listening and watching the way others have played them. Needless to say, we have deposited a great hope and expectation on it. We plan to have the first integrated by autumn 2003 and a first full working version by spring 2004.

7. ACKNOWLEDGMENTS

I would like to thank all the members of the Interactive Systems Team, and specially Martin Kaltenbrunner, for their suggestions on this paper, and Xavier Serra for his support on this project.

8. REFERENCES

[1] Agnula homepage: http://www.agnula.org

[2] Amatriain, X et al. "CLAM: An OO Framework for Developing Audio and Music Applications", Proceedings of 17th Annual ACM Conference on Object-Oriented Programming, Systems, Languages and Applications. Seattle, WA, USA, 2002.

[3] Amatriain, X., P. Arumi, and M. Ramírez, "CLAM, Yet Another Library for Audio and Music Processing?", Proceedings of 17th Annual ACM Conference on Object-Oriented Programming, Systems, Languages and Applications. Seattle, WA, USA, 2002.

[4] Arkaos homepage : http://www.arkaos.net

[5] Audiopad homepage: http://tangible.media.mit.edu/projects/Audiopad/Audiopad.htm

[6] Audiovisualizers homepage: http://audiovisualizers.com/

[7] Augmented Groove Homepage: http://www.csl.sony.co.jp/~poup/research/agroove/

[8] Barbosa, A., 2002. *Computer-Supported Cooperative Work for Music Applications*. Ph.D. Thesis proposal, Pompeu Fabra University, Barcelona.

[9] Bischoff, J., Gold, R., and Horton, J. 1978. "Music for an interactive Network of Computers". Computer Music Journal, Vol. 2, No.3, pp 24-29.

[10] Blaine, T. and T. Perkis, "Jam-O-Drum, A Study in Interaction Design," Proceedings of the ACM DIS 2000 Conference, ACM Press, NY, August 2000.

[11] Blaine, T. and C. Forlines, "Jam-O-World: Evolution of the Jam-O-drum Multi-player Musical Controller into the Jam-O-Whirl Gaming Interface," Proceedings of the 2002 Conference on New Interfaces for Musical Expression (NIME-02), Dublin Ireland, May 2002.

[12] Bongers, B. "The Use of Active Tactile and Force Feedback in Timbre Controlling Musical Instruments." *Proceedings of the 1994 International Computer Music Conference.* San Francisco: International Computer Music Association, pp. 171-174.

[13] Camurri, A., M. Ricchetti and R. Trocca, "EyesWeb - toward gesture and affect recognition in dance/music interactive systems", Proceedings IEEE Multimedia Sustems '99.

[14] Camurri, A. et al., "A real-time platform for interactive performance", *Proceedings of the 2000 International Computer Music Conference.* San Francisco: International Computer Music Association.

[15] Camurri, A., G.De Poli, M.Leman, G.Volpe (2001) A Multi-layered Conceptual Framework for Expressive Gesture Applications, Proc. Intl MOSART Workshop, Barcelona, Nov. 2001.

[16] Camurri, A., R.Trocca, G.Volpe, 2002. Interactive Systems Design: A KANSEI-based Approach, Proc. NIME2002, Dublin, May 2002.

[17] CLAM Homepage: http://www.iua.upf.es/mtg/clam/

[18] Cook, P., "Principles for Designing Computer Music Controllers", Proceedings of the 2001 New Instruments for Musical Expression Workshop. Seattle: CHI 2001.

[19] Ctugha homepage http://www.afn.org/~cthugha/index.html

[20] Eyesweb Home page: http://www.eyesweb.org/

[21] Feller, R. "FMOL Trio: Live at Metronom." Computer Music Journal, Vol. 26, No.2., pp 110-112.

[22] FMOL Home page: http://www.iua.upf.es/~sergi/FMOL

[23] FMOL-DQ Compositions Database: http://teatredigital.fib.upc.es/dq/eng/fmol/database.htm.

[24] FMOL Trio. 2000. *Live at Metronom* (CD Audio). Barcelona: Hazard Records 010.

[25] FMOL Trio. 2002. *Night in the Chapel* (CD Audio). Barcelona: Hazard Records 025.

[26] FMOL Trio. 2002. *The Köln Concert* (CD Audio). Barcelona: Hazard Records 028.

[27] Föllmer, G. Soft Music. 2001. San Francisco Museum of Modern Art; ZKM – Center for Art and Media – Karlsruhe. http://www.sfmona.org/crossfade

[28] FreeJ Homepage: http://freej.dyne.org/

[29] Gianetti, C., *Marcel.lí Antúnez Roca: Performances,Objects and Drawings*, MECAD, Barcelona (1998).

[30] Gillespie, B. 1999. "Haptic Manipulation". In P. Cook, eds. *Music Cognition and Computerized Sound: An Introduction to Psychoacoustics*. Cambridge, Massachusetts: MIT Press, pp. 247-260.

[31] Intel ® OpenCV libraries for Computer Vision and IPL libraries for image processing: http://www.intel.com/research/mrl/research/opencv/ and http://sourceforge.net/projects/opencvlibrary

[32] Ishii, H and B. Ullmer, "Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms", Proceedings of CHI 97 Conference on Human Factors in Computing systems, Atlanta, Georgia USA, 22-27 March 1997.

[33] JamODrum homepage: http://www.jamodrum.net/

[34] Jordà, S., "EPIZOO: Cruelty and Gratuitousness in Real Virtuality." Proceedings of 5CYBERCONF, fifth International Conference on Cyberspace (1996), at http://www.telefonica.es/fat/ejorda.html.

[35] Jordà, S., Aguilar, T. 1998. "A graphical and net oriented approach to interactive sonic composition and real-time synthesis for low cost computer systems." Digital Audio Effects Workshop Proceedings, pp. 207-210.

[36] Jordà, S. and La Fura dels Baus. 1998. F@ust 3.0 - FMOL (CD Audio). Madrid: Fundación Autor 02801.

[37] Jordà, S. 1999. "Faust music On Line: An Approach to Real-Time Collective Composition on the Internet." Leonardo Music Journal, Vol. 9, pp. 5-12.

[38] Jordà, S., Wüst, O. 2001. "Architectural Overview of a System for Collaborative Music Composition Over the Web." Proceedings of the 2001 International Computer Music Conference. International Computer Music Association.

[39] Jordà, S. "Improvising with Computers: A personal Survey (1989-2001)." Proceedings of the 2001 International Computer Music Conference. San Francisco: International Computer Music Association.

[40] Jordà, S. 2002. "Improvising with Computers: A personal Survey (1989-2001)." Journal of New Music Research, Vol. 31, No.1, pp. 1-10.

[41] Jordà, S. 2002. "FMOL: Toward User-Friendly, Sophisticated New Musical Instruments." Computer Music Journal, Vol. 26, No.3. pp 23-39.

[42] Klein, A. B., Colour-Music: The Art of Light. Crosby, Lockwood & Son, London, 1927.

[43] Levin, G. 2000. "Painterly Interfaces for Audiovisual Performance", Master Thesis, Massachusets Institute of Technology, pp. 53-55. http://acg.media.mit.edu/people/golan/thesis/

[44] Moritz, M., "The Dream of Color Music, And Machines That Made it Possible", Animation World Magazine, Issue 2.1, April 1997. http://www.awn.com/mag/issue2.1/articles/moritz2.1.htm l

[45] Lozano-Hemmer, R. 1996. "Perverting Technological Correctness". Leonardo, Vol. 29, No. 1.

[46] Paradiso, J. & Hsiao, K., "Musical Trinkets: New Pieces to Play", SIGGRAPH 2000 Conference Abstracts and Applications, ACM Press, NY, July 2000, p. 90.

[47] Poupyrev, I, Augmented Groove: Collaborative Jamming in Augmented Reality, ACM SIGGRAPH 2000 Conference Abstracts and Applications, p. 77.

[48] Resolume homepage: http://www.resolume.com

[49] SmallFish homepage: http://hosting.zkm.de/wmuench/small_fish

[50] Tangible Media Group: http://tangible.media.mit.edu/index.html

[51] Various Authors, 2002. Computer Music Journal companion CD. Computer Music Journal, Vol. 26, No. 4.

[52] VJCentral homepage: http://www.vjcentral.com

[53] Wessel, D. and M. Wright. "Problems and Prospects for Intimate Musical Control of Computers." Proceedings of the 2001 New Instruments for Musical Expression