THE HORGIE: COLLABORATIVE ONLINE SYNTHESIZER

Jorge Herrera

Center for Computer Research in Music and Acoustics
Stanford, USA
jorgeh@ccrma.stanford.edu

ABSTRACT

The Horgie is a new musical instrument that explores the usage of a web browser as a tool for collaborative music performance online. All the sounds are synthesized by an Actionscript application and the instrument can be simultaneously controlled on real-time by two performers, each one using a separate browser. Therefore, remote creation and performance of music is now possible without installing any other application beside a web browser with a Flash plug-in (which is already installed in the vast majority of personal computers). Special emphasis was made in the user interface in order to enable musicians and non musicians to play the instrument.

1. INTRODUCTION

Collaborative music creation and performance is a broad and interesting topic to explore. It has social, aesthetic and technical issues to be considered. There are many ways to include a collaborative aspect in music performance. Furthermore, the Internet offers the possibility of remote real-time collaboration. But how music creation and performance can take advantage of this possibility is not clear yet. This does not mean that no work has been done. In fact, there are some remarkable projects to mention.

The SoundWIRE group at Stanford's Center for Computer Research in Music and Acoustics (CCRMA) is "concerned with the use of Internet networks as an extension to computer music performance, composition and research"[1]. This group has focused on the transmission of uncompressed high quality audio through the Internet, enabling remote locations to perform tele concerts, using any kind of instrument or sound source. The results achieved are quite impressive, but they require high bandwidth connections and some applications to be installed at the endpoints.

Psychosynth [3] is another initiative focused on collaborative music creation and/or performance. It is a software synthesizer inspired by the ideas of the Reactable [6]. This project provide communication between remote synths over a network, but it requires to install the software in the remote locations. In that sense, it's not an on-line instrument.

Other projects, such as PSOs [1], have developed online instruments (instruments that run on a web browser). Mainly, this kind of projects have focused on the user interface, but have not addressed this collaborative aspect.

Another initiative worth mentioning is FMOL (Faust Music On Line) [5]. A thorough review of the different projects and applications in this field can be found at [2].

The present work proposes a different approach to the use of the Internet for collaborative music making. The idea is to provide a means for music expression using only a browser (no other installation required). All the previously mentioned initiatives require the installation of one or more applications on the machine(s) to be used. This fact impose a barrier that may discourage the potential users. Because of the same reason, the users tend to be only persons that are familiar with computer music. PSO [1] addresses this issue by providing a web browser instrument, but the collaboration aspect is handled using a central server to synthesize the sound. The purpose of The Horgie is to provide an easy to use, installation free synthesizer with real-time collaboration. The goal is to allow trained and non trained computer musicians to remotely create a sonic experience. Hopefully this will open computer music to a massive new audience. In summary, the main contributions of this work are:

- Provide and installation-free synthesizer
- Include real-time remote collaboration to the sound creation process
- Provide a sound synthesis environment to a wide audience, without the need of previous computer music knowledge

2. THE SOUND

Unlike other collaborative instruments, The Horgie synthesize the sounds on the client itself. This is consistent with the idea of no installation required. The only data transferred between clients is control data. Therefore, both can listen the exact same sounds.

In this first version, two kinds of synthesis are provided: Frequency Modulation and Granular Synthesis. Furthermore,

[1] http://ccrma.stanford.edu/groups/soundwire/

sound can also be modified by using combinations of time domain IR and FIR filters.

2.1. FM Synthesis

FM [4] is a very well know synthesis. In this case, it was implemented using phase modulation:

$$y[n] = \sin(2\pi f_c n + M \sin(2\pi f_m n)) \quad (1)$$

where f_c is the *carrier frequency*, f_m is the *modulation frequency* and M is the *index of modulation*. As will be discussed later in section 4, equation 1 is relevant not only to the sound synthesis but also to the user interface.

2.2. Granular Synthesis

The granular synthesis take a sample and extract small grains of sound that are played repeatedly. Control parameters are:

- *Inter Onset Time (IOT)*: separation between the onset of two simultaneous grains.
- *IOT randomness*: Allow to randomize the IOT, between 0 and $2 \times IOT$.
- *Grain Duration (GD)*: Length of the grain.
- *Grain Duration randomness*: Allow to randomize the GD, between 0 and $2 \times GD$.
- *Grain Envelope*: Envelope of the grain (trapezoidal or raised cosine).
- *Sample position*: Position in the sample from where the grain is extracted.

2.3. Filters

As part of the project, the Synthesis Tool Kit[2] has been partially ported to Actionscript. This partial port, which consists mainly in the *Filter* class and some of it subclasses such as *Delay* and *BiQuad*, enables the implementation of IR and FIR filters.

3. INTER-BROWSER COMMUNICATION

The collaborative aspect of the instrument is based on *Stratus* and *Real Time Media Flow Protocol (RTMFP)* developed by Adobe, which allows the direct communication between 2 clients (Actionscript applications). As stated in their site[3], "... the most important features of RTMFP include low latency, end-to-end peering capability, security and scalability. These properties make RTMFP especially well suited for developing real-time collaboration applications ... "

Using this technology, two remote performers can share the same instrument on real time (with a delay that depends on the communication bandwidth and distance between the two clients).

Two kinds of interaction are possible. The most obvious one is that any change made by one of the performers will be immediately perceived by the other (visually and audibly). But aside from that, a "chat" is included in the instrument, so the performers can share ideas and opinions in real time.

4. THE USER INTERFACE

Although difficult and challenging, implementation of real time sound synthesis and inter-client communication using a web browser is worthless if the interface used is not adequate for the task. Therefore, a major effort needs to be done in order to define and implement a compelling user interface (UI) for the instrument.

There are some important questions to answer before designing a UI. Who is the audience? What kind of experience should the user feel (should it feel like a game or like a more formal instrument)? What is the satisfaction that the application will provide?

Some design guidelines that help to answer these questions are:

- The instrument should be performable by any user. No prior music knowledge or experience is required
- Although it is not a game in the competitive sense, it should have a game-like appearance to invite the audience to play with it
- The satisfaction is the sound generated by the instrument, along with the experience of collaborative music making

An important aspect of the UI is the usability. The heuristics proposed in [7] were used to improve the usability of the system.

At the moment, four UIs have been developed and tested. The first one is based on the PD[4] or Max/MSP[5] paradigm, where different modules are connected using "wires" (see figure 1). This kind of interface is very familiar for the computer music community, but is not intuitive enough for people outside this field. Therefore, a second UI concept was developed (see figure 2). In this case, the idea is to stack modules. This paradigm is much more familiar to people outside the computer music field. In this second UI concept, there was a clear separation between the modules stacked by each performer (one of the performers stack modules on the right side of the screen while the other stack he(her)'s

[2] http://ccrma.stanford.edu/software/stk/

[3] http://labs.adobe.com/wiki/index.php/Stratus

[4] http://puredata.info/

[5] http://www.cycling74.com/products/max5

modules on the left side of the screen. Although more intuitive, this interface gives the impression of two separate instruments, which is not the concept behind The Horgie. A third UI was developed to address the issues that arose from the previous ones. It was based on the same idea of stacking sound modules, but now all of them are stacked in one pile.

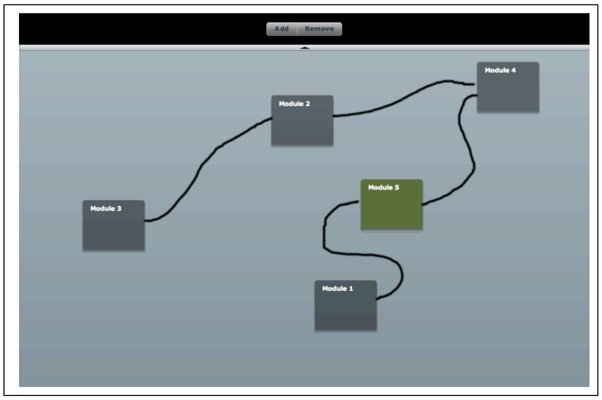


Figure 1. First User Interface

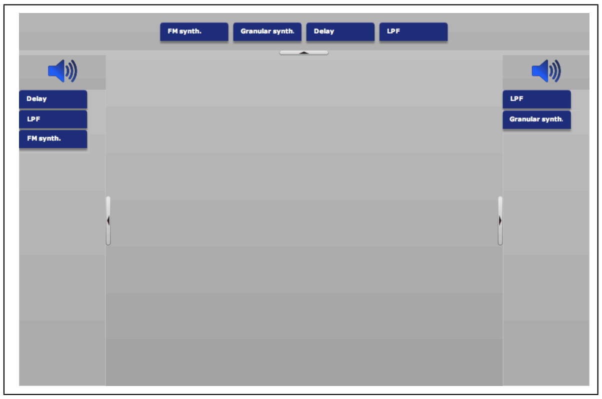


Figure 2. Second User Interface

A major improvement in the third UI was that the interface of each synthesizer was designed to abstractly represent the sound being generated by that module. For FM synthesizer, the waveform (see equation 1) is presented on the screen, so the performers can see what is being played. Also, the shape of the waveform can be modified by the performers. This kind of interface proved to be more appealing than the traditional faders plus knobs UI that is commonly used in digital instruments. In figure 3 it is possible to see the third UI developed for the case of a granulator. In this case, the circles represent grains of sound. Some visual parameters are mapped to sound parameters. For example, if the IOT is increased (less density of grains), then less circles are presented on the screen. Likewise, if the grain duration is randomized, then the size of the circles is randomized to. On the right side is the communication panel (where the performers can ”chat” while playing the instrument). The idea is to provide a familiar interface to share ideas while performing. This interface was more appealing but was complicated to understand. Also, a major drawback was that only one module can be seen at a time. Therefore, if the remote performer is modifying something in one module but the local performer is visualizing another module, the sound will change but there is no visual alert of that for the local performer.

An element included in this interface was a ”Control Point” (the red dot that can be seen in figures 3 and 4) that enables the performers to interact with the visualization of the sound. Depending on the synthesizer being controlled, the interaction with the control point changes. In the FM synth the control point is movable. The X axis controls the carrier frequency and the y axis controls the overall volume of the synth. On the granular synth, the point is not movable, but it size can be changed, controlling the duration of the sound grains.

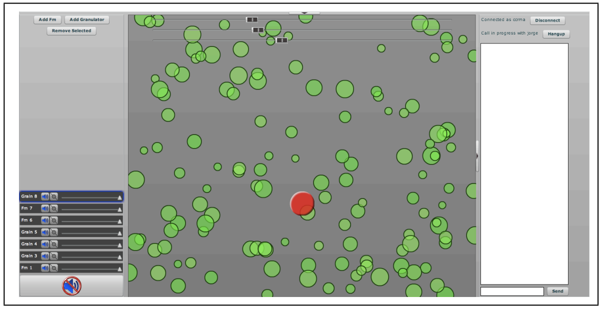
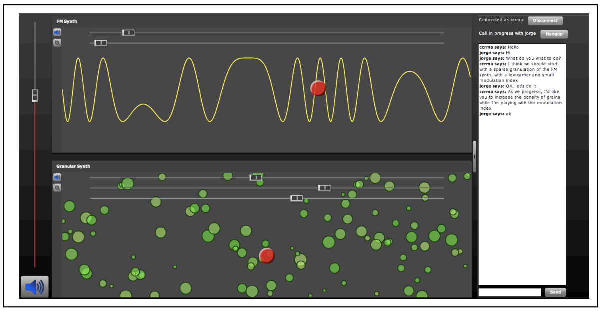


Figure 3. Third User Interface

To address this issues, a fourth UI was developed. Although the system developed allow dynamic creation of different modules, it was decided to constraint this possibility for this UI. Instead of giving freedom to the performers to create a synthesizer chain at will, a fixed setup is presented to the performers: one FM synthesizer connected to a granular synthesizer. Although this seems to restrict lots of possibilities, it was found that the interface was much more easy to understand. An interesting finding was that this highly constrained system allowed different testers to create a great variety of sounds despite the restrictions imposed. This UI can be seen in figure 4. Another benefit of this interface is that both modules are visible at all times, so visual and audio changes are always perceptible in both endpoints. This interface meets al the design requirements previously stated.

5. CONCLUSION AND FUTURE WORK

The Horgie provides a new way of collaborative sound creation using the Internet, without the need to install any software at all. It allows to create a wide variety of interesting sounds in an intuitive and easy way. There are lots of improvements and new developments to be made. Some of



Connected as ccrma

Call in progress with jorge

ccrma says: Hello
jorge says: Hi
jorge says: What do you wnat to do?
ccrma says: I think we should start with a sparse granulation of the FM synth, with a low carrier and small modulation index
jorge says: OK, let's do it
ccrma says: As we progress, I'd like you to increase the density of grains while I'm playing with the modulation index
jorge says: ok

Figure 4. Fourth User Interface

them are:

- Changes in the UI to allow more flexibility in the module interconnection without sacrificing simplicity
- Add the possibility to store and load settings
- Add the possibility to render a sound (export as WAV or MP3)
- Add the possibility to include more performers simultaneously
- Include more filters
- Develop new ways of controlling the parameters of synthesizers and filters. The idea of the control points can be extended. For example, it could be possible to change the shape of the control point and map that to some parameter of the synth.
- Develop a plug-in architecture, to allow third party developers to create visual interactions to be used
- Develop a plug-in architecture, to allow third party developers to create audio effects to be applied

One of the main problems found was the Actionscript imposes some restrictions that increase the latency of the application. In particular, the minimum buffer size allowed is of 2048 samples and it has a fixed sampling rate of 44100 samples/second. In this sense, an interesting path to follow is the exploration of new technologies that are currently emerging to improve the capabilities of web applications. It is worth to mention Native Client [8], which appears to be a really good possibility to improve the sound and visual capabilities of this kind of applications.

6. REFERENCES

[1] A. Barbosa, "Public sound objects: a shared environment for networked music practice on the web," *Organised Sound*, vol. 10, no. 3, pp. 233–242, 2005.

[2] ——, "Displaced soundscapes: Computer-supported cooperative work for music applications," Ph.D. dissertation, Universitat Pompeu Fabra, Departamento de Tecnologia, April 2006.

[3] J. Bolivar, "Psycocynth," January 2009. [Online]. Available: http://www.psychosynth.com/index.php/Main_Page

[4] J. Chowning, "The synthesis of complex audio spectra by means of frequency modulation," *Journal of the Audio Engineering Society*, vol. 21, no. 7, 1973.

[5] S. Jordà, "Faust music on line (fmol): An appoach to real-time collective composition on the internet," *Leonardo Music Journal*, vol. 9, no. 7, pp. 5–12, 1999.

[6] M. Kaltenbrunner, S. Jordà, G. Geiger, and M. Alonso, "The reactable*: A collaborative musical instrument," in *Enabling Technologies: Infrastructure for Collaborative Enterprises, 2006. WETICE '06. 15th IEEE International Workshops on*, June 2006, pp. 406–411.

[7] J. Nielsen and R. Molich, "Heuristic evaluation of user interfaces," in *CHI '90: Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM Press, 1990, pp. 249–256.

[8] B. Yee, D. Sehr, G. Dardyk, J. B. Chen, R. Muth, T. Ormandy, S. Okasaka, N. Narula, and N. Fullagar, "Native client: A sandbox for portable, untrusted x86 native code," in *2009 IEEE Symposium on Security and Privacy*, in-press.