

Wanna form a virtual band?

Fredrik Olofsson Musikproduktion

e-mail: f@fredrikolofsson.com

web: <http://fredrikolofsson.com>

Keywords

Web jamming, Networked music, OpenSound Control, SuperCollider, Virtual Instruments.

Abstract

This article describes a simple and easy to use framework for web jamming using the OpenSound Control (OSC) protocol [1] and SuperCollider by James McCartney [2]. The word "Web jamming" refers to playing music together over the Internet. This kind of musical collaboration over networks has been around for a while and has already been implemented in a number of applications [3]. Even though much of the recent development in this field is very impressive and provides the players with all sorts of customisation for sound and graphics, I still feel some very important things are lacking. Most of all I would like to see personality reflected throughout the system, and by providing a framework suitable for connecting musicians together rather than a fixed set of musical instruments for them to use, I think this can be achieved. So, by handing over the complete design of the virtual instruments, what parameters to control, how they will sound, what they will look like etc., to the players themselves, I wish to open up web jamming for individual and creative use to all kinds of musicians. If the players can express their "sound" more freely, this can compensate somewhat for delayed reactions, lack of eye contact, gestures and atmospheres lost due to physical displacement and technological limitations.

Background

I have never appreciated being forced to use someone else's musical instruments. I see it as a necessity to, so to say, bring your own instruments that you have learned to master, perhaps built yourself or just those that you like a lot. Being limited to a predefined selection of instruments where control parameters and the Graphical User Interfaces (GUI) are made by someone else or being constrained to a set of prerecorded material (such as loops) to manipulate in different ways, is not enough for me to get really hooked.

So first of all, defining your own instruments is important because some basic things are lost in a situation where the members of the band are at different physical locations. Meeting up at the rehearsal space, getting eye contact, breathing together, seeing intentional gestures etc. are examples of behaviors which can only ever be represented crudely in the virtual environment. So for a successful and enjoyable jam, I think it is very important to compensate somewhat for this by making sure everyone is feeling comfortable with their instruments and that they can contribute to the music in their own personal way. The goal is that all members' musical voices should be clearly distinguishable and personal so that true jamming can take place.

Secondly, I would like to hear new music and try out novel ways of playing with this new medium. Lots of web jamming applications are designed from a traditional way of thinking about music and the way people participate in bands and ensembles. It does not mean you have to discard all these traditional functions e.g. drummer, bass player etc., but if one for instance would like to build a master mixer and use it as one's instrument it should not be a problem. Or one person could be jamming with a noise generator and a reverb effect while another person provides the rhythm pattern, a delay effect and filters out some frequencies from the noise before routing the signal back to the first person's reverb. I see this kind of collaborative effort to create and play instruments as highly interesting and great fun. Another fascinating technique to experiment with in this medium is letting the resulting music come out in different versions and not sound exactly the same for all the members of the band!

A third feature I miss is that everybody should be able to create their own visual representation of the other band members activity. Of course, as with instruments, you should design your own functional GUI so your fellow band members can graphically monitor your playing. This is crucial as it is the only visual feedback the band is getting from you. But if someone is not satisfied with your monitoring GUI, he or she ought to have the possibility to change it or to map the input data of your activity to another GUI. So even here there could exist a great deal of discrepancies between the participating musicians' interfaces.

Taking it even further, sound and visual feedback could be totally different and highly customized for all members - yet you play, create meaning and interact together. An extreme way of jamming like this would be to transfer both representational data (coming from the other members) and your playing interface into different domains. For example, a virtual band could consist of one person sitting at home in front of a computer playing with animated shapes, one physical installation open for many participants (hardware input and feedback) and one laptop musician performing ambient soundscapes at a gig. All "members" could then be interconnected and on different locations of the planet at the same time.

Implementation

While trying to implement the features above I have chosen to work with OSC, SuperCollider and a Java UDP server. OSC is an easy to use network protocol specially designed for music and SuperCollider could be described as a modular synthesis programming language. I started out by writing a program called Samband which utilises OSC and Supercollider. Together with some foreign friends I then formed a virtual band to play with this program for fun, at clubs and at different workshops. In practice, jamming with Samband works in the following way. We decide on a time and a [virtual] place to meet up (usually some kind of simple chat service) to exchange our computers' current IP numbers. To set up the connections, the members of the band manually edit their copy of Samband to match the list of IPs excluding their own and then we are ready to start. How the music will evolve can be discussed or directed from a chat client. If it is a concert situation we also need one player at the physical location or, as a fancier alternative, a passive computer listening to and playing back what comes through the wires.

Samband is built to create a peer-to-peer network structure with no central server hence it can be used independently by any number of users and there is no need for server maintenance and hosting. An identical set of instruments reside on all connected computers and the only distributed data is control information from the users playing their instruments. This technique gives great quality of sound and very low latency compared to applications where the actual sound is compressed and streamed.

The drawbacks with Samband are that setting up and keeping connections alive is not very user friendly. Furthermore, there is a limitation on the number of parameters for each instrument, and there is no support for user designed GUIs. It is also quite hard to include and distribute new instruments made by the users. These issues are now addressed as Samband has evolved to a more all-round set of extension classes for SuperCollider. Still relying on the OSC protocol, the connection process is now made a lot easier with a different network structure in combination with a Java server. Also the restriction of how many sound parameters each virtual instrument may use is removed, there is a custom instrument GUI support plus a few handy tools like nicknames, server status report, a scan for lowest CPU and smallest screen size, global tempo and key information.

Still left to solve is automatic distribution of code, sound file support and issues concerning internal signal routing, timing and synchronization of actions. I also believe that common use in different contexts will reveal many new problems and possibilities for this framework to deal with.

On the accompanying CD-ROM is a simplified version of this framework with some demo instruments to try out. If properly installed, one can login and with some luck find other readers connected at the same time to start jamming with. Please read the documentation and follow the installation instructions carefully.

For more information, downloadable source-code and current status of this project, direct your browser to <http://www.fredrikolofsson.com/webjamming>

References (as of Oct. 2002)

- [1] SuperCollider <http://www.audiosynth.com>
- [2] OpenSound Control <http://cnmat.berkeley.edu/OSC>
- [3] a few examples:
<http://www.livejam.com>
<http://www.keyworx.org>
<http://www.transjam.com>