FEEDBOXES

Krzysztof Cybulski
ul. Klopotowskiego 4/11
03-717 Warsaw, Poland
kcybulski@pangenerator.com

ABSTRACT

Feedboxes are interactive sound objects that generate rhythmic and harmonic patterns. Their purpose is to create intuitive tools for live improvisation, without the need for using computer with midi controller or fixed playback. Their only means of communication is sound - they "listen" with the microphone and "speak" with the speaker, thus interaction with Feedboxes is very similar to playing with real musicians. The boxes could be used together with any instrument, or on their own – in this case they create a feedback loop by listening and responding to each other, creating ever-changing rhythmic structures.

Feedboxes react to incoming sounds in simple, predefined manner. Yet, when used together, their behaviour may become quite complex. Each of two boxes has its own sound and set of simple rules.

Author Keywords

NIME, Generative music, Physical computing, Interactive device, Feedboxes, PdDroidParty, Pure Data.

ACM Classification

Applied computing - Sound and music computing, Human-centered computing - Sound-based input / output

1. INTRODUCTION

In a search for intuitive ways of creating improvised electronic music, I was looking for a solution to free the musicians from the constraints of typical laptot setup, which would be a laptop, midi controller and PA system. This situation creates a metaphor, in which an interface for interaction (i.e. the actual physical actions, like turning knobs and pressing pads) is detatched from sound generating device (the computer) and the sound source (PA). Additionally, performers' eyes are often focused not on the interface (which would be natural in case of traditional instrument), but on the computer screen. This situation is very different from the familiar experience of using an acoustic instrument, where the interface, sound generating device and sound source are the same object. Also, most often musicians use multiple virtual instruments (or one istrument with many sounds or tracks) in one computer, which further complicates the process.

This is a problem not only from the point of view of the performer, but also for audience members – there is no clear correlation between the observed action and the music that is heard.

One of the possible solutions is to create a separate device for each virtual instrument, and to combine all of the above functions (interface, sound generating and sound projection) in one device.

NIME'17, May 15-19, 2016, Aalborg University Copenhagen, Denmark.

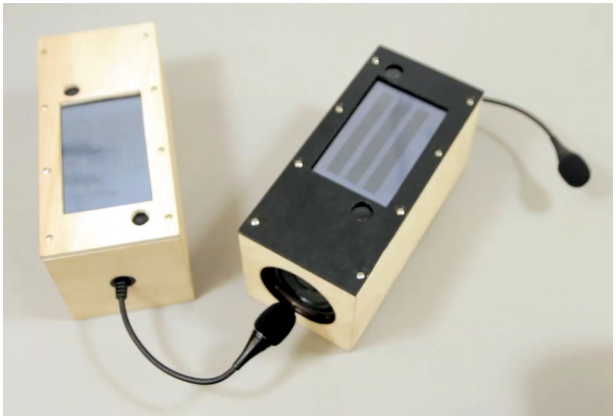

Figure 1. Feedboxes

This solution is the basis of Feedboxes. Additionally, Feedboxes are "autonomous" devices, which means thei're not controlled directly, but rather they act in a way an improvising musician does – they respond to incoming sound by generating another sounds on their own. Apart from simple user interface, used for setting up basic parameters (sensitivity, threshold, decay etc.) the interaction is executed with sound itself and by changing respective physical positions of Feedboxes.

2. TECHNICAL DETAILS

Each of two Feedboxes comprises of simple Android smartphone (Telefunken Sierra) which runs custom Pure Data patch with the help of PdDroidParty [1] – an application written by Chris Mccormick, based on libpd library. PdDroidParty is a kind of wrapper app, which makes it easy to run Pd Vanilla patches on any Android device after slight modification.

The smartphone is mounted inside a small wooden box, together with small battery-powered speaker and gooseneck microphone. With fully charged batteries, the box is wireless and can be handled freely.

Each of two boxes runs a different Pure Data patch, with different sounds and different ways of reacting to incoming sounds:

2.1. Rhythm box

Rhythm box generates semi-random beat, the complexity of which depends on the volume of incoming audio signal. There are two variables that could be changed with varying volume of incoming sound:

- Density of the beat, which is achieved by switching between eight available preprogrammed three-track eight-note sequences. First sequence contains no notes, second

contains only basic half-note beat, and so on, up to eighth sequence which is a dense beat.

- Pitch of samples

2.2. String box

String box generates plucked or strummed 4-note chords (made with Karplus-Strong synthesis) based on short, percussive sounds (i.e. clapping, beat generated by Rhythm box). Quieter sounds pluck all notes at once, louder sounds strum one note after another. Playing or singing long notes (longer than a second) allows for changing the harmonic content of the chord. The sound appears with certain delay, which can be adjusted to one bar length of the rhythm box – this way Feedboxes can play together in the same tempo.

3. SCENARIOS FOR INTERACTION

3.1 Feedboxes on their own

As stated earlier, Feedboxes are controlled with sound. Thus, they can also interact with each other by the means of sound – they only need initial trigger (for example clapping, which usually will trigger the String box, or some longer sound, which would rather trigger Rhythm box). The boxes need to be situated in a way that they could listen to one another, with the microphone of one box close to the speaker of the other box (fig. 2)

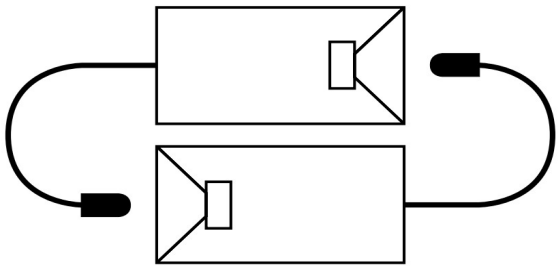

Figure 2. Feedboxes in a feedback loop

When one of the boxes starts to play, the other one is triggered by its sounds and also starts to play. The relative positions of the boxes change the volume of sound captured by the microphones, thus the response of the boxes changes. The closer the mic, the denser the beat or louder the chords/strums, and vice versa. When a stable configuration is found, boxes may be left on their own to play slightly variating structures.

3.2 Feedboxes with another instrument or voice

Feedboxes might be also situated "in series" with another instrument, played by live musician: (fig.3)

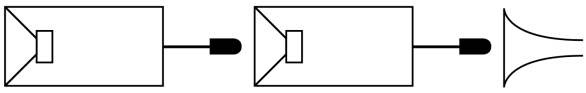

Figure 3. Feedboxes with another instrument (trumpet)

In this configuration, Feedboxes play only when there is a sound coming from the live instrument (for example trumpet). When a melodic instrument is used, the preferred order would be: instrument → Rhythm box → String box. With percussive instrument (for example xylophone), the order would be: instrument –> String box → Rhythm box.

This configuration allows free form improvisation for the instrumentalist, with interactive accompaniment by the boxes. With melodic instrument, it's also possible to change notes played by String box during the performance, by playing longer notes (more than 1 second). The live istrument should be then directed more towards the microphone of String box (percussive sounds of Rhythm box could interrupt the process of setting up new chordal notes.

4. CONCLUSION

Feedboxes, still being a new project (completed in december 2016), have been tested and proved to be a responsive tool for improvisation. They were outlined in an article written by Peter Kirn for Create Digital Music blog [2]. Most of all, they're a lot of fun to play with, which was one of intended outcomes. Typical performance scenarios involving digital technology, such as playing with fixed playback, or controlling the software with midi controllers, are somehow detaching from the very act of creating music, especially for musicians that feel at home with their instruments of choice. Feedboxes provide a way of intuitive interaction with digitally generated sounds, which follow the musician, at the same time providing some harmonic or rhythmic constraints in a similar fashion that live improvising musicians would. Thus Feedboxes provide an alternative way of using digital technology in the context of improvised music.

5. REFERENCES

[1] C. Mccormick. PdDroidParty. 2013. http://droidparty.net

[2] P. Kirn. FEEDBOXES are autonomous sound toys that play along with you. Create Digital Music, December 16, 2016 http://cdm.link/2016/12/feedboxes-autonomous-sound-toys-play/

6. Appendix

Video documentation of Feedboxes: https://vimeo.com/195930695

Feedboxes on authors' website:
http://krzysztofcybulski.com/feedboxes.html