Designing a Virtual Orchestra for Real-Time Interaction

Joaquín Jiménez
Universidad Pompeu Fabra
1932 Wallamaloo Lane
Wallamaloo, New Zealand
joaquinhernan01.jimenez@estudiant
.upf.edu

ABSTRACT

This article describes the ideas behind the design, structure, and use of the patch presented during the final concert for Real-Time Interaction Class. It shows the justification for the project and decisions made during the design of the internal mechanism, programming and the design of the user interface. Also, included in this article is the explanation of the philosophy and ideas that support the final version of this project.

Author Keywords

NIME, Pure Data, Real-Time, interaction, computer, music

1. CONCEPT

The birth of the electronic musician brought many new paradigms to the craft of performing music live. We cannot avoid comparison with traditional ways of playing music. ¿Is this an instrument? ¿Is it music performance? ¿Is is playing live? ¿Is the performer a musician? ¿Does, he/her, need musical training?

A proper answer to the previous questions goes beyond the scope of this document but lays the ground to define the idea behind this patch. The patch is intended to be a machine to perform music, without being a traditional instrument or needing a score. It is instead, a modular arrangement of devices, which leave the user the task of performing or interacting with it, as a conductor in the sense and context of electronic music performance with electronic instruments. In this system, the user not necessarily a musician or trained in music, but can appreciate a musical composition, can express and understand the performance of musical ideas. I simplified the interaction with the system to be played using physical controllers consisting of knobs and buttons, an interface that is common to many of the users.

The patch is designed to be a consistent tool with the functions of an orchestra inside, or in some way carry the role of such a group of Digital Music Instruments in the same package and using the same interface. The concept of this patch is a structure that creates music using interaction with a performer, and the performance generates a new level of abstraction that is above the individual control of notes.

Another essential characteristic of this patch is to be able to be reprogrammed before every performance. Even when this tool is a machine that produces music, the program can change each time we perform with it. The design of this patch sets flexibility as a high priority aspect of it. Another important consideration is to give the performer the ability to improvise with it. A set of parameters are open to being manipulated by the user during real-time execution, giving him the possibility and power to express and provide different performances in the same context. The freedom is given by the open parameters to improvise, but the design doesn't allow the user to go beyond the area defined by the patch, in other words, it gives a context definition, and the user can express freely within this piece.

The structure doesn't allow the user to create without interaction, in other words, the patch doesn't perform by itself. If the user doesn't provide interaction events, nothing musical can happen. This way, the piece guarantees that the user is creating all musical and creative outcome.

Visual feedback is necessary for instruments. All real instruments provide this kind of information to performers. The fact that we can program a patch allows us to provide ad-hoc feedback devices and let know the performers what is going on inside a particular device or the whole patch.

2. INSPIRATIONS/STATE OF THE ART

Machines today have the power to free the performers from the standards of musical execution (in the context of musicians in a group playing notes on instruments while reading or remembering a musical score). If the performer wants to follow a score or improvise with it, computers can support either decision. If we know a performing task is repetitive or needs a lot of attention, we better leave it to an automatic system, which is more precise and can follow instructions tirelessly, moving the creative work to the performer.

Computers are programmable, we have been able to create interactive software we depend on with full trust or confidence, so why not use it for creative tasks?

This patch, in particular, is inspired by modular systems, which work using electric signals and separate functions by modules. We can interchange and make copies of this modules to duplicate tasks and create different versions or just use them for various purposes. Modular systems have been proved to present an excellent platform for music creation, performance, and real-time composition.

3. FIRST ITERATION AND PERFORMANCE

During the course, we were asked to create an instrument, but first without live performance in mind. To use it during the first concert it was necessary to adapt it to a real-time interaction system. In particular, to synchronize it with other elements, to use the same interface and mapping to a controller.

Mapping was straightforward once the modules were adapted. Even so, better rehearse time would have improved the performance. I think this is good because that means that the performance we created was not depending on the patch, it depended on the ability and knowledge of the performers.

4. SECOND ITERATION AND PERFORMANCE

For the second concert, I included characteristics that allowed the modules to be integrated with the others, allowing for the independent development of modules that are compatible amongst themselves, with the possibility to define the higher abstractions of the patch in later stages.

It was easier to start again with a new patch during the second iteration, now that I knew how to use the modules in the second concert. I applied the technique of designing the interface first and the underlying mechanisms later, this way it was easier for me to think about the performance avoiding limiting the functions to what the instrument allowed me to do.

After developing the modules, the integration was more natural than the first iteration. I didn't need to change the modules during integration at all. Performance wise, it was more direct to control modules that were created to play along. I knew the capabilities and limitations of each one and didn't have to re-learn or practice each one now that all follow the same design patterns.

5. STRUCTURE

The patch contains two types of modules. Control module (Sequencer) and instruments (Granulator, Kick, Snare, and Drone):

Sequencer: It can save up to 5 different sequences generated or modified by the user, real-time or pre-recorded. The user can create this sequences by drawing notes using the computer's mouse. The tempo division of each step in the sequencer allows for values 1/2, 1/4, 1/8, and 1/16. This module sends out three values, continuous (to use for modulations), discrete (to send notes), and Bang, to trigger events every time the sequence sends the highest value, for example, drums or percussion.

Input: Sequence number, Start, Stop, Time Division.

Output: Continuous value, Discrete Value, Bang

Granulator: This is an instrument that granulates the soundwave drawn by the user in the GUI. Internal LFOs control the play head that granulates the soundwave. The sliders below change the LFO behavior.

Input: (On the GUI) Scrub Speed, Range, Position, LPF Freq. (From a Sequencer) sets the position for scrubbing the soundwave.

Output: Audio to DSP.

Kick: This module synthesizes a kick drum sound and responds to bangs sent by a Sequencer. Contains parameters to change the envelope and filter of the sound.

Input: (On the GUI) decay, release, resonance, low pass filter and volume. (From a Sequencer) Bang.

Output: Audio to DSP.

Snare: This module synthesizes a snare sound and responds to bangs sent by a sequencer. Contains parameters to change the pitch, delay and amount of noise.

Input: (On the GUI) pitch, repetition, delay and noise. (From a sequencer) Bang.

Output: Audio to DSP.

Drone: Creates a drone sound based on white noise, modulations, and phasing.

Input: (On the GUI) modulations and phases. (From a sequencer) continuous value to set low pass filter frequency.

Output: Audio to DSP.

6. LESSONS LEARNED

Many conclusions may be derived from this performance, but the most important are:

- A good performance is achieved better with rehearsals; the performer should not rely on the quality of the patch.
- It is better to design modules with live performance in mind (when the intention is to create a live performance)
- Design before building a patch.

7. FUTURE WORK

Time constraints were an essential factor in the creation of this patch and performance. Without this limitation, I can produce a better sound design. Also, I would improve the visual feedback to take better decisions when performing.

I would not use this patch for professional work, but it is an excellent way to test and sketch new ideas.

8. EVALUATION AND DIFFERENCES BETWEEN FIRST AND SECOND ITERATIONS

I (or we) presented different patches with different goals in mind during first and second concerts. The first patch contains elements that were not designed to play together in a real-time situation, neither for mapping to OSC Touch, and it was evident. The second time, the patch and instruments were designed specifically for real-time interaction in mind. The characteristics, advantages, and disadvantages of the physical controllers were mapped directly into the patch the second time.

For me, the second performance was technically and sonically better than the first and reflects what we learned in class. I don't agree with the students grading the outcome of this patch. To be both judge and jury (in one own's case), when doing this for the first time doesn't make sense to me and can result in misleading grades or losing focus on the class objectives. Also, the evaluation of aspects like musical quality, virtuosity, sonic and musical possibilities makes me want to think about what is the purpose, or what we should have learned in this class; I am still confused about it. In my opinion, the instructions for this concert have to be more specific about the grading concepts and skills that we need to show in the patch and during the performance.

9. REFERENCES

[1] Eigenfeld, A. Real-time Composition or Computer Improvisation? A composer's search for intelligent tools in interactive computer music. Electroacoustic Music Studies Network. 2007.

[2] Collins, N. Towards Autonomous Agents for Live Computer Music: Real-time Machine Listening and Interactive Music Systems (Doctoral Dissertation). Centre for Music and Science Faculty of Music. University of Cambridge. 2006.

[3] Winkler, T. Composing Interactive Music: Techniques and Ideas Using Max. MIT Press, 2001.

[4] Jordà, S. Interactivity and live computer music.