ESMERIL - An interactive audio player and composition system for collaborative experimental music netlabels

Cristiano Severo Figueiró
NuSom
ECA/USP
São Paulo, Brasil
figocris@gmail.com

Bruno Rohde
CAHL
UFRB
Cachoeira, Bahia, Brasil
brunorohde@gmail.com

Guilherme Rafael Soares
CECULT
UFRB
Santo Amaro, Bahia, Brasil
organismo@gmail.com

ABSTRACT

ESMERIL is an application developed for Android with a toolchain based on Puredata and OpenFrameworks (with Ofelia library). The application enables music creation in a specific expanded format: four separate mono tracks, each one able to manipulate up to eight audio samples per channel. It works also as a performance instrument that stimulates collaborative remixings from compositions of scored interaction gestures called "scenes".

The interface also aims to be a platform to exchange those sample packs as artistic releases, a format similar to the popular idea of an "album", but prepared to those four channel packs of samples and scores of interaction. It uses an adaptive audio slicing mechanism and it is based on interaction design for multi-touch screen features. A timing sequencer enhances the interaction between pre-set sequences (the "scenes") and screen manipulation scratching, expanding and moving graphic sound waves. This paper describes the graphical interface features, some development decisions up to now and perspectives to its continuity.

Author Keywords

Interactive Design, Granular Synthesis, Sampling, Mobile Applications, Free Software, NetLabels

CCS Concepts

• **Human-centered computing → Auditory feedback ;**
• **Human-centered computing → Open source software;**
• Human-centered computing → Collaborative interaction ;
• Human-centered computing → Touch screens ;
• Human-centered computing → Mobile devices ;
• **Applied computing → Sound and music computing;**
.

1. INTRODUCTION

ESMERIL is a free platform for music creation, performance and distribution - an Android mobile application that plays music in an open format and enables an interactive listening experience which we could understand as some kind of "expanded musical reproduction" of a song that will be always opened to new combinations. The application is an instrument intended for artists and enthusiasts of digital music to recompose existing materials in real time, generating new gestures and sound results, which can be recorded on the same interface. In addition it can be used as a platform for sharing music with the possibility of collaborations and remixes.

NIME'19, June 3-6, 2019, Federal University of Rio Grande do Sul, Porto Alegre, Brazil.

The application uses the idea of symmetric slicing to define loop regions. The possibility of defining loops of different sizes allows the creation of complex rhythms with non-synchronized cycles. The interface allows switching between beatslice and multigranular modes allowing overlap between sequenced and multi-touch segments directly in the waveform. The design of the interface was done in such a way as to show all possibilities of remixing on a single screen.

This project was developed in view of the experimental music NetLabels ecosystem. In these communities artists collaborate by exchanging audios and creating remixes with different versions of the same songs. The idea was to develop a platform that allows the insertion of audios created in other platforms and creative setups but that lead to a certain standardization of the format for remixing and sharing.

From the perspective of free software [5] and the enhancement of cultural and autonomous artistic practices, ESMERIL is an open platform both in code - published with GPLv3 3 - license and in the production of the content that will be made available. New "scenes", the name of the scored compositions used by ESMERIL, can be submitted to a catalog of "sample packs" by anyone, to participate in trade with a community of developers and fellow musicians or some possibility of commercialization of scenes individually.

2. OVERVIEW

ESMERIL was conceived during the preparation of the I ESME - Experimental Music Netlabels meeting, held in Salvador, Bahia, in October 2018. During the festival, the software was already presented and tested in a beta version. We held the first creative lab with artists from sixteen NetLabels from seven states of Brazil. During the meeting, artists from different music aesthetics collaborated in various ways to test the interface and create musical content for the platform by creating "scenes", bringing a collaborative character to the process, something we looked for since the beginning.

ESMERIL is developed with all its core coded in the PureData (PD) environment with the graphic user interface based on

Ofelia library, which is based on Open Frameworks (OF) toolkit. Since it is PD and LibPD [1],[2] based this software can be a multiplatform tool - it works on any operating system (with PD) and in could be compiled as binary to the most varied types of hardware, from old computers, through cell phones and tablets, to high end computers and video games.

In this first release Bruno Rohde developed and coded with the help of a research team with previous experience with PureData libraries, testing Android compiling methods and user experience of Cristiano Figueiró, Guilherme Soares and the work of Rafael de Marchi designing the frontend widgets.

3. GRAPHICAL INTERFACE FOR MOBILE DEVICES

ESMERIL has a graphical user interface that can be used as an application for Android devices, mapped for its touch screen gestures interaction and totally independent from its interactive PureData source code - which can be used direct on PD with mouse interaction and the possibility of hacking some parts of code that still not exposed in the mobile ready interface.

The mobile GUI interface allows the switching between 2 interaction modes for each channel with the possibility to control the qualities of the interaction from the functions of slicing, sample selection, waveform scratching and effects. In figure 1, in the center of the screen we see the waveforms with the column of choice of modes on the left and the column of functions on the right, and also a transport bar at the top.

Figure 1: ESMERIL GUI overview

One of the objectives of interface development was the intuitive feature, allowing users who are not intimate with audio software to have a smooth experience with the resulting sound. In this sense all the interaction is done in a single screen without hierarchical menus by directly touching the waveform alternating between two modes and selecting four different functions.

Each channel has two modes, beatslice mode and multigranular mode. Each one allows a different interaction on the waveform. These modes are derived from designs developed in the Música Móvel [4] project.

Multigranular mode offers a way to "navigate" the sound through the possibility of multitouch and scratches in the waveform. This procedure transforms the sound samples into polyphonic instruments with intuitive manipulation. The slice structure determines the size of each slice that corresponds to a grain. Each touch on the waveform triggers a loop in the selected grain.

Beatslicer mode fires loops in regions with slices. The size and position of the loops is defined in the slicing function with fine controls or directly altered in the waveform with the gesture of pinching and stretching. This loop can be altered with controls of the effects function by manipulating the order of reading the slices in random, backward, inverted and "frozen", with the possibility of overlapping the effects.

The slices can be divided in very small parts, which can be synchronized with a master BPM beat reference. The user can browse the parts, squeeze, stretch, move from and for different parts of the loop - all with the touch of the fingers.

All those gestures can be recorded in “moments” to make some kind of composition sequence of moments with changes in the size, position and mixture of sample slices, as well as the application of some presets of effects like freeze the sound block, reverse the loop play, ping pong loop and some more.

Since those gestures are recorded they can be played in a sequence with conceptually we call “scene” and works as some kind “demonstration” of the sample pack with a possible score of gestures. Ever new pack of samples that is loaded will come with a inicial preset of a scene. The user can follow the composition of the scene and interact with it or disable the sequence in the transport bar and create new moments. This allows for infinite radical re-creations with the sound samples.

However, those procedures of recording of the gestures are still not part of the mobile interface and are made in a coding compositional way with patches in PD that we will discuss from now on as “hacking methods”.

4. COMPOSITIONAL HACKING INSIDE PURE DATA

Creating and changing scenes for ESMERIL currently requires the user to install and run the source code through the main patch. Maybe in the future this process will be incorporated into the main GUI, but for now it is a way to musicians who are developing scenes to know and learn about programming int the pure data environment.

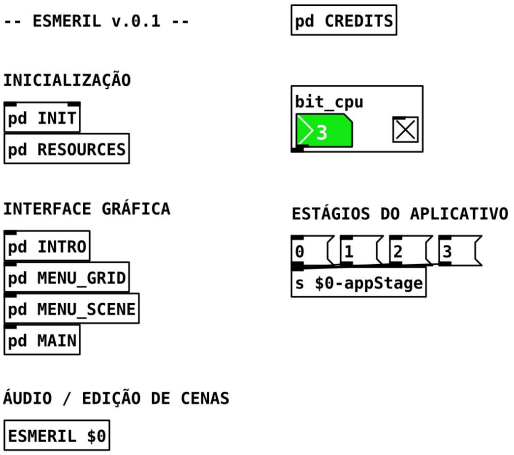

Figure 2: [main.pd] screen

Since some of the features still hardcoded into PureData patches, part of the musical gesture for composition or even for live performances is an invitation to join the core development group and learn some about how this software is being made as an open source legacy for its continuity inside netlabel communities.

4.1 Some hacking methods

So, for some of this procedures we will describe the actual methods that the core developer and beta testers are using nowadays. Anyway, we suggest that the user experience with the steps taken to have more control of this features be reported as “feature requests” as a feedback for a GUI implementation at interaction level for the next releases.

The method of hacking the ESMERIL code is to run the main patch [main.pd] with Pd vanilla version> 0.47 with the external libraries Zexy and Ofelia 1.0.8, before and apart of any process of compilation for mobile devices.

The actual hardcoded exposed patches seeks to respect best practice guidelines in Pure Data [3] in a effort to make evident, in the visual layout and comments of the code, the operations performed for each functional block and the exchange of data between them.

4.1.1 Modularity

The principle of modularity has always been used when possible, reducing programming time and facilitating further code maintenance. A set of abstractions was developed for functions performed multiple times. They are located in the "date / abs /" folder, and are named with two different prefixes, bit_ for the abstractions of the audio mechanism and parameterization system and memory, and gui_ to those related to the graphic interface.

4.1.2 Audio

The implemented audio mechanism is based in the symmetric slicer, similar to B/I/T/S/L/C [4], but with a completely new approach to code. The approach was inspired by Kelly's [6] work, and basically consists of operating multiple layers of audio players from a common phase, which makes the synchronization between them perfect and constant, even with variations in time.

From the global phase, each player can use simple mathematical operations - multiplication, division, addition, modulo - for phase manipulation and obtaining specific segments (slices or slices) for linear reproduction or following certain patterns.

This mechanism controls the slicing system and defines the quality of the interaction by the auditory result from the size of the slices.

The interaction in both modes depends on the way of slicing the samples which can be completely different in each channel allowing easy composition of complex rhythms.

4.1.3 Modular Score Memory System for Composition of the scenes

All the abstractions created for the ESMERIL that need management of multiple parameters use the COMEMO System - Control System for Modular Memory. This system makes possible to implement new control parameters during the process of developing a Pure Data application in a simple and fast way. The system consists of a set of abstractions - [bit_param], [bit_memo], [bit_memopack], [bit_memounpack], and [bit_listbox] - that work in an chained way, allowing to receive and send data of each parameter by a communication pattern, managing the view of the state of the controls on one or more objects in the interface, as well as store data in text files for later retrieval.

The "memo.txt" file stores the data of the four players of the slicer and the audio mixer at each defined time. The polyphonic granular player, on the other hand, does not store its parameters in the “scene” memory, being used only as a performance element. Each scene can contain up to 64 different moments, which are composite playback arrangements using the program's main interface in conjunction with the memory manager [bit_listbox], located in the [ESMERIL $0] patch as as seen in figure 3.

The COMEMO System synchronizes the main interface controls of the application with the interface of the abstractions [bit_track] in the patch, both of which can be used to create arrays with all the data that will be structured in the memo.txt file.

The sequence of moments executed in the scene is defined in another file called meta.txt, which stores metadata of the samples of each scene and the sequence of moments defined in memo.txt.

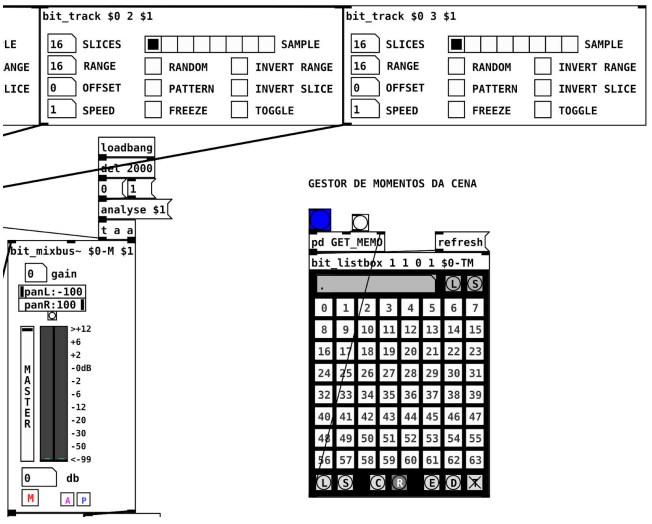

Figure 3: Screen of [ESMERIL $0] abstraction

The moment recording and scene creation system is one of the features of ESMERIL that combines precise composition and sequencing with samples and intuitive interaction directly on the waveform. In this sense we can say that ESMERIL is a hybrid instrument blurring the boundary between composition and improvisation.

As access to sound samples and compositing files is opened, in each scene the user gets in touch with the musical thinking of each artist in a more in-depth and invited to remix and collaborate.

5. FUTURE

As the system works on top of PureData it allows the inclusion of all externals functionality provided by the community in the Desktop environments. However, some of these functions are not automatically transposed to the Android build. Some important features that are in the "to do" list for the next versions of ESMERIL are:

a) A more integrated GUI system for the navigation of netlabels and artists "scenes". For now this interface is planned as a separated feature to be accessed at a web browser CMS based platform.
b) Integrate the score composition system of the "scenes" parameters and methods of gesture recording with the same interaction design GUI of the actual slicer. For now this interface is hardcoded in PureData..
c) Synchronize different devices by a common protocol like "Ableton Link" and/or similar.
d) A separated access for sound mixer and chain of sound effect layers at a different GUI windows with same interactive design standards of the slicer. This part still also hard coded in PureData.
e) OSC (Open Sound Control) interface API for mapping external physical devices control of the interface or sync with external hardware for audio and video that already uses the protocol.
f) Refactoring the GUI interface for the new version of Ofelia.. The library was redesigned for those next versions (> 2.x) and some adjustments will need to be made to fit the new Lua script based system. We could also consider other similar alternatives.

6. ACKNOWLEDGMENTS

We would like to thank to all the people who participated in various ways in building the ESMERIL network and application, including all NetLabels and collaborators. This work was only possible from the efforts of many members of the Pure Data community, and we are proud to be able to continue this work in some way. Thanks also the core PD and libraries developers Miller Puckette, Zack Lee, Ed Kelly, Peter Brinkmann, Dan Wilcox and Antoine Rousseau for generously posting their research and software used in the ESMERIL construction process.

7. REFERENCES

[1] BRINKMANN, Peter. Making musical apps. O'Reilly Media, Inc, 2012.

[2] BRINKMANN, Peter et al. Libpd: Past, Present, and Future of Embedding Pure Data. In: Proceedings of the Pure Data Convention. 2016.

[3] FARNELL, Andy. Designing Sound. The MIT Press, 2010.

[4] FIGUEIRÓ, Cristiano; ROHDE, Bruno; SOARES, Guilherme. Música Móvel Apps – Experimentation and open design for musical instruments in Android. In Proceedings of the Pure Data Convention. 2016.

[5] FREE SOFTWARE FOUNDATION. O que é Software Livre? [2017?]. Disponível em: <https://www.gnu.org/philosophy/free-sw.pt-br.html>. Acesso em: 24 jan. 2019.

[6] KELLY, Edward. Casting a Line: A Flexible Approach to Soundfile Playback Using libPd in Ninja Jamm. In: Proceedings of the Pure Data Convention. 2016.

8. LINKS

ESMERIL in action:

https://www.youtube.com/watch?v=_ctZNm8hqV4

Source code on Github:

https://github.com/brunorohde/ESMERIL

Website:

http://esmeril.ufba.br

Manual with explanations in the interface functions:

http://esmeril.ufba.br/wp-content/uploads/2019/01/esmeril_manual.pdf