Code{strata} Sonifying Software Complexity

Denez Thomas
University of Rennes 2
Rennes, 35000, France
denezthomas@gmail.com

Nicolas Harrand
INRIA
Rennes, 35000, France
nicolas.harrand@inria.fr

Benoit Baudry
KTH, Royal Institute of Technology
Stockholm, Sweden
baudry@kth.se

Bruno Bossis
University of Rennes 2
Rennes, 35000, France
bruno.bossis@univ-rennes2.fr

TEI '18, March 18–21, 2018, Stockholm, Sweden
© 2018 Copyright is held by the owner/author(s).
ACM ISBN 978-1-4503-5568-1/18/03.
https://doi.org/10.1145/3173225.3173317

Abstract

Code{strata} is an interdisciplinary collaboration between art studies researchers (Rennes 2) and computer scientists (INRIA, KTH). It is a sound installation: a computer system unit made of concrete that sits on a wooden desk. The purpose of this project is to question the opacity and simplicity of high-level interfaces used in daily gestures. It takes the form of a 3-D sonification of a full software trace that is collected when performing a copy and paste command in a simple text editor. The user may hear, through headphones, a poetic interpretation of what happens in a computer, behind the of graphical interfaces. The sentence “Copy and paste” is played back in as many pieces as there are nested functions called during the execution of the command.

Author Keywords

Copy and paste; Instrumentation; Java; Sonification; spatialization; Cycling 74 Max 7.

ACM Classification Keywords

Operational Analysis, Auditory Feedback, Media Arts.

Introduction

Software is, by nature, invisible, yet it is omnipresent in our lives. This pervasiveness of software is the result of organic and decentralized processes. A growing number

Code{strata}

Materials: The installation itself is made of concrete, and sits on a wooden board and trestles. The audio is diffused through Beyerdynamic headphones (DT-770 Pro, 80Ohm or higher), with a Raspberry Pi 3 model B and a I2S sound card (IQAudio Pi-DAC+). 5V (2A min.) USB power supply.

Preparation: The whole device must be positioned as shown in Figure 1. The concrete computer sits on the left of the desk, headphones on its right. If shipping appears to be too complicated, a new piece of concrete can be poured in situ. This installation can be exhibited either in light or dark place. In the latter case, it might be slightly illuminated with a warm white.

of companies and individuals have contributed to the development and aggregation of millions of software pieces. The assembly and coordination between all these pieces relies on a flexible hierarchical organization of software and a stratified organization of the code. Inspired by the natural wonders of structure and complexity, we want to restore a more tangible relation between users and the software world through artistic representations. We choose to focus first on the strata of *ctrl-c ctrl-v*, an iconic action that symbolizes the major impact of software on all sectors of our societies. We deconstruct all the code that has been stacked to provide this action in almost any software tool. We choose an open source text editor to pursue this deconstruction of code strata. The choice of open source code is essential for us since it provides an access to all strata, but also to the origin of these strata.

The idea behind this project is not to frighten by showing how complex software is, but to reveal the beauty of those great networks of organized data and computation. The first example that we were looking at, when we started working on the project, was a work by William Bradford Paley, entitled *CodeProfiles* [4]. To illustrate the complexity of a running software, he chose to show the lines of code. He was actually able to show an exhaustive visualization of the code, because the entire program was written in order to display itself. That is what makes our approach at opposite ends of Bradford Paley's. As we didn't want to come to a simple data visualization, we chose to build a static installation, made of wood and concrete. This paper describes the creation process and gets into the ins and outs of the installation.

A sound installation

Representing a computer system unit with a concrete block was a way for us to illustrate the opacity and the simplicity of high-level interfaces. A very strong contrast is made between what we may see, namely a fully closed and compact block, and what we can actually hear.

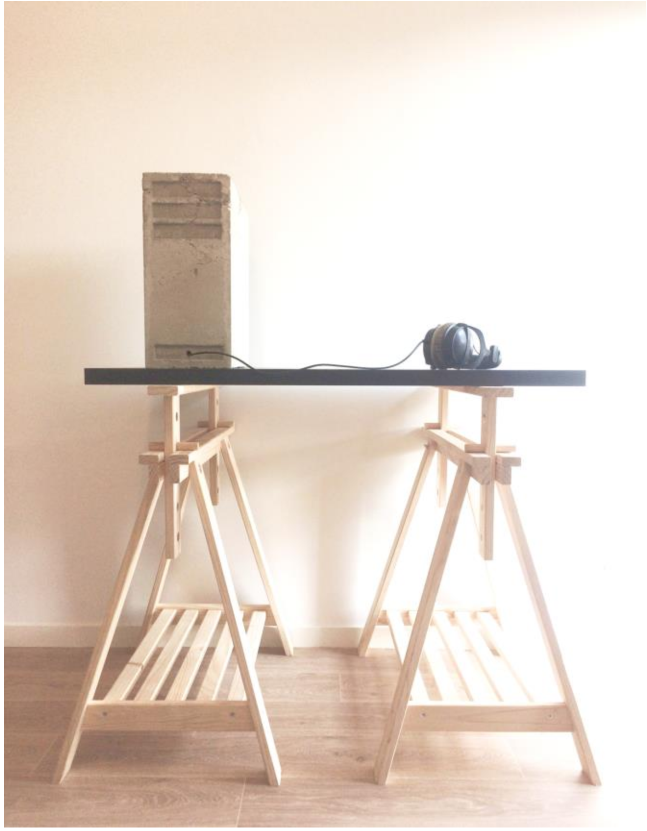


Figure 1: Full installation front view.

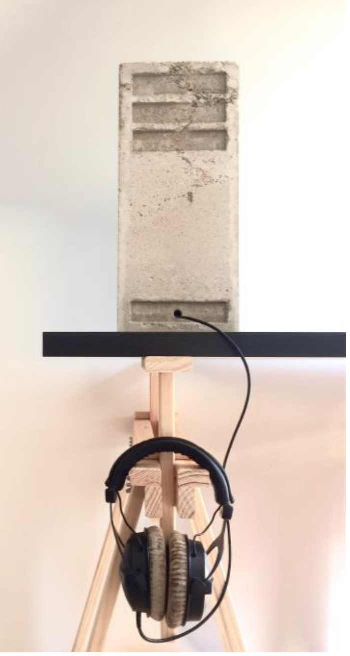


Figure 2: Closer view of the concrete computer.

The audio spatialization is meant to virtually place the listener in the center of a very wide room. The symbolic act of listening through headphones to what is happening inside the computer gives the installation all its meaning. This is why the diffusion is only possible through headphones. As if we were using a stethoscope, we can hear each individual detail in all of the complex activity. The user cannot interact with the complete loop (about 15 minutes). What we can hear may be considered as kind of sound landscape [7], in which the beginning and the end are of little relevance.

During the first exhibition of the project, a few visitors told us it was reminding them of the sound of a multitude of insects. More generally, the idea of nature has been raised a lot. The analogy with nature, and more precisely with geology has been part and parcel of the project's early stages. Imperfections that are inherent to concrete (see Figure 2) are giving the device an almost archeological aspect. It seems necessary to remember that all the functions called by the program we analyzed were someday written by humans, hence the construction materials. Wood and concrete are somewhat reminiscent of this nature-culture duality. Resting on a wooden table and trestles, the concrete block has visible foundations which are not without pointing out the concept of strata.

An audio interpretation of software data

One could consider this project as a zoom or a slow motion of what happens when a ‘copy and paste' action is performed. Nevertheless, it is important to understand that this is not a realistic or exhaustive interpretation of data. Because as soon as there is interpretation, there are choices to be made. This project intends to convey a positive and tangible image of computing complexity. In that sense, this project aims at letting human users reclaim software complexity, as a genuine postdigital artistic installation [1]. Actually, an exhaustive exploration is impossible. For instance, we cannot measure the exact time elapsed between two instructions. In fact, time is subject to exogenous variations, starting with all the instrumentation methods that have been implemented into the source code, which are drastically slowing down the whole process. Another issue that needs to be taken into account is the high concentration of activity in the mid-range levels. The first stratum we work with is the application and libraries calls (JavaFX, the software library for the graphical interface). The code and the dataset are open source[1].

In Figure 3, columns represent the strata of code execution, when rectangles are the functions (blue for the application itself and green for the JavaFX library). White lines represent the way each function is calling sub functions. If we read linearly through the whole data set, we must consider the fact that the sound, which will be created in Cycling 74's Max, will be looping a long time around the same levels. As the average visitor only listens to the sound for a few tens of seconds, the experience may be different for each one of them. Sometimes, some will hear only little or no variations during their listening, such moment should not be perceived as less relevant than another.

[1] https://github.com/DIVERSIFY-project/code-strata

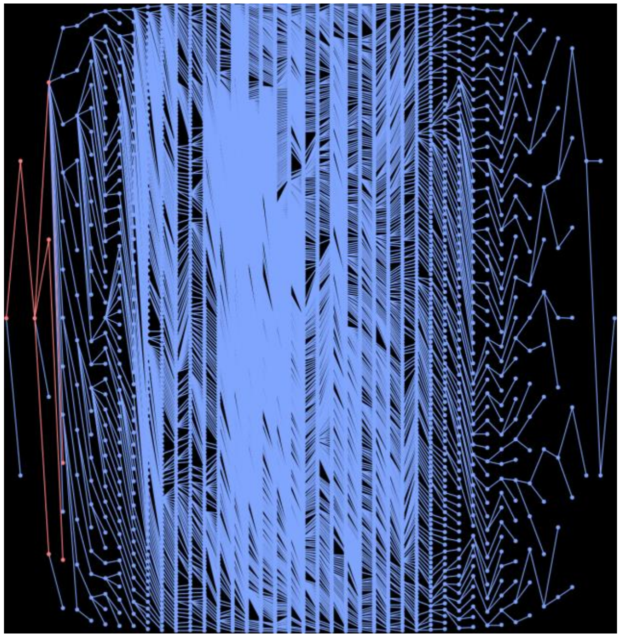
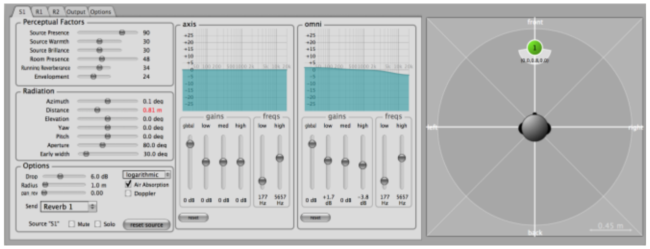


Figure 3: Application and library calls.

The use of Cycling 74’s Max programming environment [4] is particularly appropriate when dealing with interpreting data. In fact, the results of the source code instrumentation in a JSON text file can later be loaded into Max for it to be usable. Max can read through the entire data and use all elements as parameters for manipulating sounds. For now, each JSON object represents a method that has been instrumentalized, and contains a few elements describing it. The data structure must be considered as a tree, where each method necessarily has a parent, and possibly brothers and children. The first element of an object is its maximum depth, followed by its number of brothers, children, the method’s name, the weight and current absolute level. By scaling/mapping the output of each element, the values are adjusted to fit the parameters we want to modify in the audio synthesis or processing. Inspired by the work of composers like Gérard Grisey in *Les Chants de l’amour* [2], or Jonathan Harvey in *Mortuos Plango, Vivos Voco* [3], we worked on the ambiguity between human voice and computer processing.



Source Presence 90
Source Warmth 30
Source Brillance 30
Room Presence 48
Running Reverberance 34
Envelopment 24

Azimuth 0.1 deg
Distance 0.81 m
Elevation 0.0 deg
Yaw 0.0 deg
Pitch 0.0 deg
Aperture 80.0 deg
Early width 30.0 deg

Drop 6.0 dB
Radius 1.0 m
pan rev 0.00
logarithmic
☑ Air Absorption
☐ Doppler
Send Reverb 1
Source "S1" ☐ Mute ☐ Solo

axis gains: global 0 dB, low 0 dB, med 0 dB, high 0 dB; freqs: low 177 Hz, high 5657 Hz
omni gains: global 0 dB, low +1.7 dB, med 0 dB, high -3.8 dB; freqs: low 177 Hz, high 5657 Hz

Figure 4: IRCAM Spat Operator (Graphical User Interface).

We took a sample of human voice, whispering the sentence ‘copy and paste’. The whole sound is played back in as many fragments as there are nested functions called during the execution of the command. The length of each grain is indexed to the maximum depth of the function. If a method is called by many other methods, then we may hear a long grain (“cop”, “pa”, “and”). If it is called by a few other methods, then we shall hear very short grains (“t”, “k”, “p”). The position of the sample's reading head is directly linked to the reading of the tree. In our case the copy and paste takes almost fifteen minutes to be fully heard. Using human voice as raw material for the sonification of the software reminds us of all the work that has been indirectly made by so many for us to be able to perform even such simple operations. We also spatialize the sound with a binaural encoding by IRCAM's Spat [6].

The virtual distance between the audio source and the listener depends on the function's calling level. We are placed in the middle of the 45 levels, the first one being behind our head, the last one being in front of us.

References

1. David M. Berry. 2014. Post-digital humanities: Computation and cultural critique in the arts and humanities. *Educause*, *49*(3): 22-26.
2. Bruno Bossis. 2004. Les Chants de l'amour de Gérard Grisey, entre rigueur formelle et jubilation humaniste. In Danielle Cohen-Lévinas, *Le temps de l'écoute, Gérard Grisey ou la beauté des ombres sonores*. Paris, L'Harmattan, p. 229-270.
3. Bruno Bossis. 2004. *Mortuos Plango, Vivos Voco* de Jonathan Harvey ou le miroir de la spiritualité. *Musurgia*, *XI (1-2)*: 119-144.
4. William Bradford Paley. Whitney ArtPort. Retrieved November 4, 2017, from http://artport.whitney.org/commissions/codedoc/paley.shtml
5. Cycling'74. About Max. Retrieved November 4, 2017, from https://cycling74.com/products/max
6. IRCAM. Spat. Retrieved November 4, 2017, from http://forumnet.ircam.fr/product/spat-en/
7. Raymond Murray Schafer. 1993. *The Soundscape: Our Sonic Environment and the Tuning of theWorld*. Rochester, Inner Traditions Bear and Company, VT.