Web Orchestra Studio: a real-time interactive platform for music and education

Juliano Kestenberg[1*], **Vitor Rolla**[1*], **Djalma Lúcio**[1] , **Luiz Velho**[1]

[1]VISGRAF Laboratory
Instituto Nacional de Matemática Pura e Aplicada – IMPA
Estrada Dona Castorina 110, Jardim Botânico – 22460-320 – Rio de Janeiro, RJ – Brasil

julianok@impa.br, vitorgr@impa.br, dlucio@impa.br, lvelho@impa.br

Abstract

In this paper, we introduce the Web Orchestra Studio, a set of applications which enables the development of musical concerts for laptop orchestras. We offer an open-ended platform for collective artistic experimentation which can be utilized by experts or non-initiated students. In order to instance some of the platform features, we present a case study describing our participation in the Math Festival activities with the workshop Music, Mathematics and Computers. Fundamentally, with this work we intend to leverage academic debate concerning the interdisciplinary fields of music, computer science and education.

1. Introduction

In the past decade the first laptop orchestras appeared. Generally, such ensembles are formed by academic professionals who have background knowledge in audio programming and music. Usually, the orchestras are associated with a post-graduate program within an academic context. Therefore, they are considered a fruitful environment for conducting interdisciplinary research in technology and music.

In this paper, we introduce a web platform called Web Orchestra Studio. This platform was utilized to build a workshop experience tailored to children in the age range of 5-12 years old. The event took place in Rio de Janeiro in April 2017, as part of the Math Festival activities.

This paper is organized as follows. In the next section, we present the most well-documented laptop orchestras. In Section 3, we describe the Web Orchestra Studio, a set of applications that make it possible to prepare a concert and build a laptop orchestra. In Section 4, we report the workshop Music, Mathematics and Computers – presented during the Math Festival – as a case study of our platform. Finally, we expose conclusions and future work.

*Supported by CAPES.

2. Related Work

In this section, the most well-documented laptop orchestras are presented chronologically.

The first laptop orchestra was founded in 2005 at Princeton University. Its activities include presentation of basic programming concepts, individual and group assignments presentation, and rehearsal as an ensemble. PLOrk, or the Princeton Laptop Orchestra [1], uses a homogeneous collection of six-driver hemispherical loudspeakers for instrument-like acoustic dispersion.

The Carnegie Mellon Laptop Orchestra [5] (CMLO) is part of a course in computer music systems and information processing at Carnegie Mellon University. The students learn techniques for audio and MIDI [13] programming, real-time synchronization and scheduling, and music representation. At the end of the course, they must present a piece of music which is performed by the orchestra.

The Milwaukee Laptop Orchestra [7] (MiLO) grounds its practice in free improvisation. The orchestra is based on the NRCI (Networked Resources for Collaborative Improvisation) software, which was developed in Milwaukee University. The NRCI is a suite of Pure Data [3] tools. Instrumental performances and video projections are common practice among the members.

The Stanford Laptop Orchestra [8] (SLOrk) is a large-scale ensemble that explores cutting-

Name	Software	Seats	Since
Princeton Laptop Orchestra (Plork) [1]	Chuck [2], Pure Data [3] or SuperCollider [4]	15	2005
Carnegie Mellon Laptop Orchestra (CMLO) [5]	Serpent [6]	10+	2006
Milwaukee Laptop Orchestra (Milo) [7]	Pure Data	8+	2007
Stanford Laptop Orchestra (Slork) [8]	Chuck in general	20+	2008
Linux Laptop Orchestra (L2ork) at Virgina Tech [9]	Pure Data	15+	2009
The Machine Orchestra at CalArts [10]	Chuck in general	12+	2010
Birmingham Ensemble for Electroacoustic Research (BEER) [11]	SuperCollider	3-5	2011
Cybernetic Orchestra at McMaster University [12]	Chuck, Pure Data or SuperCollider	10+	2012
Web Orchestra Studio	HTML5 + CSS + JavaScript + Python	5+	2017

Table 1: Laptop Orchestras

edge technology in music. It provides a platform for research in instrument and sound design, as well as in music composition and performance. The orchestra also offers a classroom environment which combines music, technology, and live coding [14, 15] performance.

The Linux Laptop Orchestra [9] (L2ork) at Virgina Tech relies mainly on Pure Data for audio, video, and graphics processing. They use Nintendo Wiimotes and built-in laptop input devices (e.g. keyboard, track-pad, webcam) as instruments. They also utilize external soundcard combined with custom-built hemispherical speakers for audio output. The ensemble infrastructure currently supports up to 15 fully networked performers.

The Machine Orchestra [10] at the California Institute of the Arts is a mixed ensemble of humans and robotic performers. Its pedagogical focus is to provide the necessary knowledge to create a robotic instrument or to control the set of robotic instruments previously built for the orchestra.

The pedagogical nature of an orchestra is explored by David Ogborn [12] and Scott Wilson et al. [11]. Ogborn presents the laptop orchestra [12] from University of McMaster, Canada. Wilson depicts the Birmingham music group of electroacoustic research [12], United Kingdom. Both explore the participatory aspects involved in a group of musicians and their laptops, promoting shared responsibility in the development of new pieces, and inclusive atmosphere based on peer learning.

The motivations for starting a laptop orchestra, both in musical and cultural terms, and its aesthetic and technical issues can be verified in [16]. An excellent discussion on composing for laptop orchestras is available in [17]. Different strategies for sound design, conduction roles and improvisation are also addressed in the aforementioned paper.

3. Web Orchestra Studio

Web Orchestra Studio (WOS) is a set of applications that enables the development of musical concerts for laptop orchestras. The platform comprises the following elements: Composer, Concert, Maestro, Performer and Server. All applications were developed in HTML5, CSS and JavaScript, except the Server – developed in Python language. The communication between the different laptops occurs through websockets [18]. WOS allows one to create a complete concert, from its conception to its deployment in a number of laptops, making them ready for presentation.

The concert creation and deployment process

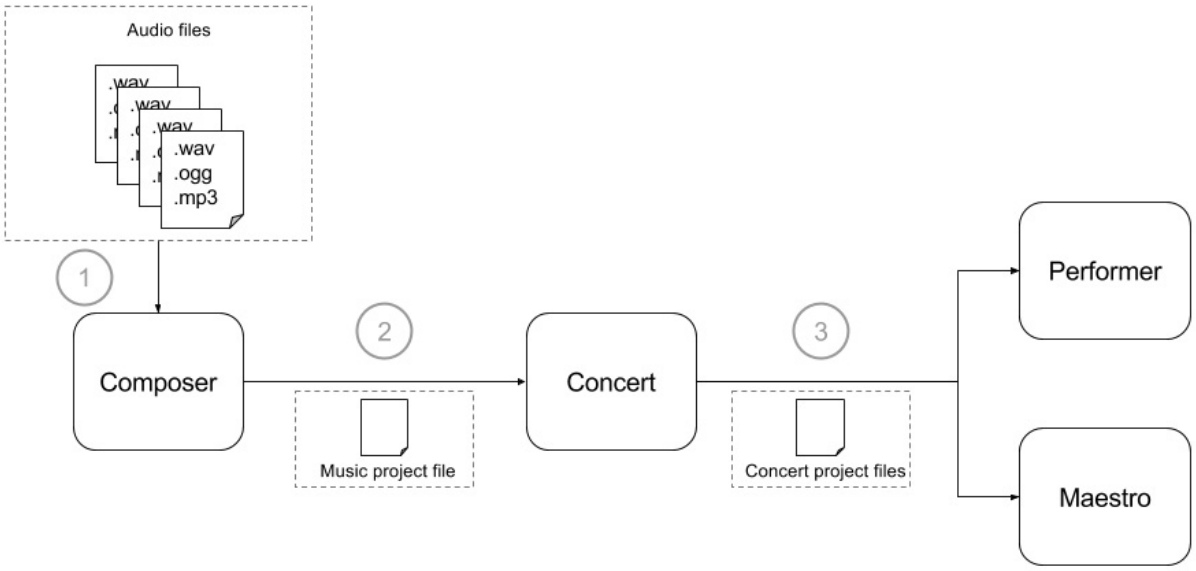

Figure 1: Pipeline for musical concert creation and deployment

is accomplished by following the pipeline depicted in Figure 1:

1. The user composing a musical piece selects the audio files that will be utilized in the composition;
2. In order to define which audio files will be copied to each instrument, a music project file is created using Composer application;
3. Finally, the Concert application generates the concert project files that shall be deployed in the Performer and Maestro applications.

Each application is detailed below:

3.1. Composer

The Composer application is responsible for song creation. It stores audio samples in the following formats: WAV, Ogg and MP3. Each song can be played by up to 20 instruments. An instrument can be a computer, a smartphone or a tablet. Each one can play several samples at once.

3.2. Concert

The Concert application determines the order in which each music will be played. Then, it forwards the concert project files to the laptops running Maestro and Performer applications.

3.3. Performer

The Performer application loads the project created in Concert so that each music can be played as defined in Composer. Each instrument configured in Composer is associated with a single Performer.

3.4. Maestro

The Maestro application loads the project created in Concert and controls which music will be played in the Performer applications. This component interface allows the conductor to see which instrument is being played and how long it will take for its sound to be ceased. It is also possible to interrupt the sound that is being played in the Performer applications anytime.

3.5. Server

The Server application is responsible for the communication between Performer and Maestro. Commands sent by Maestro are received by Server and forwarded to every Performer in the network. Reversibly, Performer contacts Maestro through Server. The Server operates with real-time communication.

4. Case Study

In this section, we present a case study which illustrates a fraction of our platform features. The workshop Music, Mathematics and Computers has been the selected event. It took place in Rio de Janeiro in April 2017 as part of the Math Festival activities.

Figure 2: Laptop orchestra performers making eye contact with the conductor

The workshop main goal was to sensitize students and the general public to the beauty of the relationships between music, mathematics and computers. In order to achieve this goal, we have explored several connections between musical structures, mathematical concepts and computer implementation methods. Our challenge was to provide a rousing atmosphere that could inspire children and teenagers in their learning paths.

The workshop was divided in two main parts. First, the audience was introduced to basic concepts presented along three sections: Introduction to sound and music; Music and Math; and Music with the computer. This theoretical part lasted about 30min. Subsequently, our laptop orchestra was set up, with volunteers – drafted from the audience – playing musical pieces which were previously chosen by our team. The second part took about 20min. Our workshop was given during the four days of the Math Festival. We had a total number of 12 sessions.

4.1. Setup

Our laptop orchestra comprised five computers used by performers and one destined to the conductor. Each performer station had the following components:

1. an HP laptop running Linux Ubuntu 16.04 operating system;
2. a small desk on which the computer rested;
3. a mono-directional speaker for sound amplification;
4. a cushion on which the performer sat.

The five stations were arranged so that the ensemble drew an imaginary semi-circumference on stage, with performers facing the conductor, whose laptop rested on a transparent pulpit. A wired audio network connected computers with a multi-channel audio interface.

4.2. Preparation

After introducing the fundamental concepts, we invited people from the audience to play the computer meta-instruments. Mostly, their ages ranged from 5 to 12 years old.

As the players took seats, we explained the activity dynamics. Keyboard space bar was the only key which would make the computer deliver sound. Performers were asked to make eye contact with the conductor so they could play according to his manual gestures (Figure 2). They

were also made aware that each musical piece would have its corresponding background color on the laptop screen.

4.3. Action

Four were the musical pieces performed by our young orchestras:

1. C Major arpeggio;
2. Reproduction of a five-note melody extracted from the movie "Close Encounters of the Third Kind", directed by Steven Spielberg and released in 1977;
3. Interpretation of *Pontos de Lagrange*, composed by Brazilian computer music researcher Marcelo Cicconet;
4. Interpretation of *Cadência Universal*, also composed by Marcelo Cicconet.

C Major arpeggio served as a warming-up in which performers had the chance to experience the system responsiveness for the first time. Players took turns and delivered a single note, either Dó, Mi or Sol.

In order to reinforce the concepts of rhythm and musical tempo, which had been presented previously, the five-note movie melody was reproduced twice. First, with a slow tempo induced by the conductor. Then, with a fast tempo and notes with a shorter duration.

Both musics composed by Marcelo Cicconet had the purpose of exercising improvisation. Each performer delivered a melodic instrument sample, ranging from piano to electric guitar, to saxophone. The samples were meticulously prepared so they could sound harmonious in multiple combinations.

At first, the conductor was a member of our system conception team. Towards the end of each performance we invited children from the audience to conduct the orchestra. That was a memorable experience as it granted us with the opportunity to observe the system running without our direct control.

4.4. Feedback

After each session, Math and Physics teachers approached our team asking for extra information regarding the concepts which had been introduced. Notably, the teachers expressed their interest in reproducing the workshop in their local learning environments. Hence, they asked for supportive digital material and technical guidance. Finally, children were delighted for having had the opportunity to take part in our laptop orchestra (Figure 3). They shared their thoughts on how to improve the overall performance.

Figure 3: A laptop orchestra portrayed in the Math Festival

5. Conclusions and Future Work

Laptop orchestras constitute a generous learning environment to expand the overlapping fields of music, computer science, and live performance. With Web Orchestra Studio we offer an open-ended platform for collective artistic experimentation which can be utilized in different instruction levels.

The workshop described in the previous section represents an instantiation of the Web Orchestra Studio. Particularly, we intended to aid children learning quests by awakening playful ways of experiencing mathematics through music and computers.

Our project next steps include the following tasks: (i) make the software suite available for free download; (ii) conclude the project web portal; and (iii) encourage the utilization of the platform as a tool for music and math education.

Ultimately, we believe in the power of combining computers capabilities with the uniqueness of human responses. Web Orchestra Studio is simply one of the countless ways to explore new education paradigms through computer-mediated technology.

References

[1] Ge Wang, Dan Trueman, Scott Smallwood, and Perry R Cook. The laptop orchestra as classroom. *Computer Music Journal*, 32(1):26–37, 2008.

[2] Ge Wang, Perry R Cook, et al. Chuck: A concurrent, on-the-fly, audio programming language. In *Proceedings of the 2003 International Computer Music Conference*, 2003.

[3] Miller Puckette et al. Pure data: another integrated computer music environment. *Proceedings of the second inter-college computer music concerts*, pages 37–41, 1996.

[4] James McCartney. Supercollider: a new real time synthesis language. In *Proceedings of the 1996 International Computer Music Conference*, 1996.

[5] Roger Dannenberg, Sofia Cavaco, and B. Aygun J. Baek E. Barndollar D. Duterte J. Grafton R. Hunter C. Jackson U. Kurokawa D. Makuck T. Mierzejewski M. Rivera D. Torres A. Yu E. Ang, I. Avramovic. The carnegie mellon laptop orchestra. In *Proceedings of the 2007 International Computer Music Conference, vol II*, pages 340–343. The International Computer Music Association, 2007.

[6] Roger B Dannenberg. A language for interactive audio applications. In *Proceedings of the 2002 International Computer Music Conference*, pages 509–515, 2002.

[7] Christopher Burns and Greg Surges. Nrci: Software tools for laptop ensemble. In *Proceedings of the 2008 International Computer Music Conference*, 2008.

[8] Ge Wang, Nicholas J Bryan, Jieun Oh, and Robert Hamilton. Stanford laptop orchestra (slork). In *Proceedings of the 2009 International Computer Music Conference*, 2009.

[9] Ivika Bukvic, Thomas Martin, Eric Standley, and Michael Matthews. Introducing l2ork: Linux laptop orchestra. In *International Conference on New Interfaces for Musical Expression*, pages 170–173, 2010.

[10] Ajay Kapur, Michael Darling, Dimitri Diakopoulos, Jim W Murphy, Jordan Hochenbaum, Owen Vallis, and Curtis Bahn. The machine orchestra: An ensemble of human laptop performers and robotic musical instruments. *Computer Music Journal*, 35(4):49–63, 2011.

[11] Scott Wilson, Norah Lorway, Rosalyn Coull, Konstantinos Vasilakos, and Tim Moyers. Free as in beer: Some explorations into structured improvisation using networked live-coding systems. *Computer Music Journal*, 38(1):54–64, 2014.

[12] David Ogborn. Live coding in a scalable, participatory laptop orchestra. *Computer Music Journal*, 38(1):17–30, 2014.

[13] Gareth Loy. Musicians make a standard: the midi phenomenon. *Computer Music Journal*, 9(4):8–26, 1985.

[14] Nick Collins, Alex McLean, Julian Rohrhuber, and Adrian Ward. Live coding in laptop performance. *Org. Sound*, 8(3):321–330, December 2003.

[15] Marc J Rubin. The effectiveness of live-coding to teach introductory programming. In *Proceeding of the 44th ACM technical symposium on Computer science education*, pages 651–656. ACM, 2013.

[16] Dan Trueman. Why a laptop orchestra? *Organised Sound*, 12(02):171–179, 2007.

[17] Scott Smallwood, Dan Trueman, Perry R Cook, and Ge Wang. Composing for laptop orchestra. *Computer Music Journal*, 32(1):9–25, 2008.

[18] I. Fette and A. Melnikov. The websocket protocol. RFC 6455, RFC Editor, December 2011.

Posters