Soundworks – A playground for artists and developers to create collaborative mobile web performances

Sébastien Robaszkiewicz
IRCAM – Centre Pompidou, STMS lab
IRCAM-CNRS-UPMC, Paris
sebastien.robaszkiewicz@ircam.fr

Norbert Schnell
IRCAM – Centre Pompidou, STMS lab
IRCAM-CNRS-UPMC, Paris
norbert.schnell@ircam.fr

ABSTRACT

We present *Soundworks*, a Javascript framework that enables artists and developers to create collaborative music performances where a group of participants distributed in space use their smartphones to generate sound and light through touch and motion. The framework adopts a modular architecture to make it easy to implement different performance scenarios, using a server / client architecture supported by `Node.js` and `socket.io`. We provide a boilerplate that allows anyone to bootstrap a scenario for *Soundworks* and focus on its audiovisual and interaction design instead of the infrastructure. We developed three scenarios based on *Soundworks*: *Wandering Sound*, *Drops*, and *Paths*. Each scenario has been tested multiple times with teenagers during workshops at the Centre Pompidou's Studio 13/16 in Paris. We present the results of these live tests.

Categories and Subject Descriptors

H.5.3 [**Information Interfaces and Presentation**]: Group and Organization Interfaces – Web-based interaction; H.5.5 [**Information Interfaces and Presentation**]: Sound and Music Computing – Systems

Keywords

music, mobile music, HTML5, Web Audio API, collective instrument, collaborative performance, interaction design, Node.js, modular architecture, Javascript framework

1. INTRODUCTION

Recent improvements in mobile technologies made it possible for anyone to have a device with multiple built-in sensors in their pockets. Artists took advantage of this technology to enable new forms of musical expression in the smartphones [8, 2] and turn them into self-contained instruments [10]. In parallel, some research focused on the use of distributed computing devices to make music, such as the laptop orchestras [9]. Taking advantage of the miniaturization of the components and of the phones' mobility and ubiquity, further research explored setups that involve a larger number of users equipped with mobile devices. For instance, Wang's and Essl's *Mobile Phone Orchestras* [11, 5] let the electronic music ensemble members perform for an audience, whereas Lee & Freeman's *echobo* [4] invites the audience to play simplified mobile phone-based instruments that follow a chord progression determined by the artist and broadcasted over a local network. While solo performances, music ensembles, and participative concerts based on mobile technologies usually feature trained performers, mobile devices also conquered multimedia installations in which technology supports the audience's autonomous interactions [6].

With the massive adoption of smartphones and the recent extensions of web technologies, large-scale collaborative music is more than ever within reach of anyone. The recent HTML 5 APIs now provide support for a wide range of sensors on the phones, and the Web Audio API allows to control very finely the audio on the web. Consequently, users now only need a simple URL to turn their smartphones into musical instruments, which dramatically lowers the barriers to entry to digital music experiences.

With *Soundworks*, we intend to provide a framework for developers and artists to create participative musical experiences distributed across the audience: by simply connecting to a webpage on their mobile browsers, players will be able to take part in a collaborative[1] performance where not only the sound comes from the smartphones for a more immersive experience, but also where the audience has an active role to play to influence the sound and music.

2. THE SOUNDWORKS FRAMEWORK

2.1 Description

Soundworks is a framework that enables artists and developers to create collaborative performances in which participants are distributed in space, and use their smartphones to create sound and light through touch and motion. The framework allows for placing the participants on any arbitrary topology of seats (*e.g.* in a theatre) to take advantage of their positions in the global audiovisual rendering.

As an example, the players in one side of the room could propagate a sound wave to the other side of the room through the distributed smartphone loudspeakers if they shake their smartphones at the same time.

2.2 Architecture and implementation

In order to connect the smartphones with each other, we implemented a client-server architecture using a `Node.js`

[1]While music performances are usually *collaborative* as soon as multiple performers play together, here the term collaborative rather applies to the technical infrastructure (i.e. the instrument) used in the performance.

1st Web Audio Conference (WAC), January 2015, Paris, France.

server and the `socket.io` library to pass messages between the server and the client. Since we want *Soundworks* to be used by as many artists and developers as possible, we opted for a modular architecture. For instance, some scenarios may require the smartphones to be beat-synchronized with each other, while others may use the same type of web audio synthesizers, and others yet may include a global visualization on a screen. We want each of these components to be easily reusable to help artists and developers kickstart their projects and share their developments with others.

Scenarios based on *Soundworks* share a common logic and architecture. For instance, the way the server handles the client connections and the place assignment (*i.e.* where each smartphone is physically) is a shared setup process. Consequently, we created a client-server infrastructure that people can build upon. Both client and server sides are divided into two major parts: *setup* and *performance*.

When the players connect to the server (*i.e.* by opening a given URL in their browser), they may be guided through a *setup* process before they are ready to play. As an example, they may have to get or choose their place in the room, synchronize their devices so that their smartphone shares the same beat as the others, and calibrate the compass. Each setup module is autonomous: when the setup of a module is done, the module sends a `done` event to the global `Setup` module. Then, when all setup modules of a particular player are `done`, the global `Setup` module (on the server side) emits a `ready` event: the player finished the *setup* process and can join the second phase of the scenario, the *performance*.

With this architecture, the *setup* process of each player can be easily composed from an arbitrary set of predefined and/or user-defined modules (cf. Figure 1). That way, any artist or developer who wants to create a scenario on *Soundworks* can start from an existing example based on predefined modules and concentrate on the sound and interaction design of the *performance* without taking care of the infrastructure and standard setup procedures.

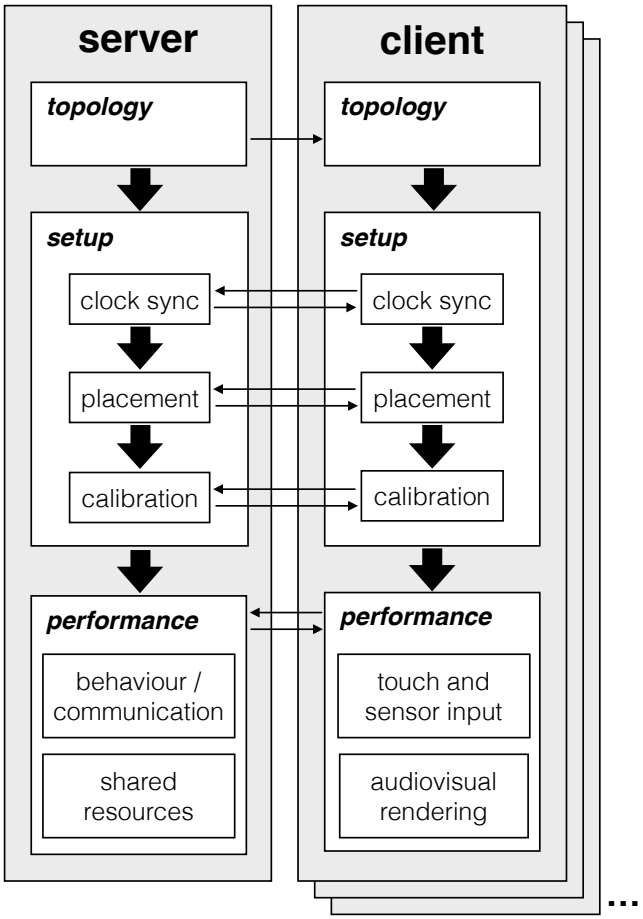

Figure 1: *Soundworks*' modules runtime sequence (vertical arrows) and communication between server and clients. In this example the setup consists of three stages, clock synchronisation, player placement, and a calibration (*e.g.* the compass).

3. SCENARIO EXAMPLES TESTED WITH REAL USERS

We tested the concept and implementation of *Soundworks* from the very beginning in a series of workshops (the *Collective Sound Checks*) in collaboration with the Studio 13/16 at the Centre Pompidou in Paris. Three of them took place in May and June 2014, and four of them between October and December 2014. Each workshop lasted 4 hours: teenagers were free to join and leave the activities at their will, and they usually stayed between 15 and 90 minutes.

During these workshops, we invited groups of teenagers to perform several *Soundworks*-based scenarios. Each teenager was provided with an iPod Touch (5^{th} generation) and a portable loudspeaker (cf. Figure 2). Each performance lasted about 10 minutes, during which the teenagers could play and engage with the scenario.

3.1 Wandering Sound

The first scenario we implemented with *Soundworks* is *Wandering Sound*. A representation of the spatial topology of the participants — usually a matrix — appears on the screen of a *player* who becomes a *soloist*. By moving a finger on the matrix on the screen, a soloist controls which smartphones emit light and sound. The *soloist* can choose between a set of sounds generated with the Web Audio API, which are modulated by the speed of the finger on the screen. This way, the *soloist* remotely plays on the other *players*' smartphones and makes the light and sound wander across the room. After some time, another *player* becomes *soloist* and takes over the control of light and sound.

During the *Collective Sound Checks*, we invited groups of teenagers to perform *Wandering Sound* sitting on a grid of 4×3 foam pads on the ground. Many participants found that is was interesting and enjoyable to play as soloist on an instrument that involved a large space and other players. On the other hand, the teenagers reported that they wanted to experiment with more sounds, and have less "waiting" time when they are not *soloists*. For instance, this suggested scenarios where the players who are not soloists can modulate the sound emitted by their smartphone. However, after a few sessions, we realised that the interaction scenario of *Wandering Sound* was strongly based on the idea of a single performer playing on the smartphones of the others distributed over space without taking sufficiently into account the interactions between the players. This encouraged us to develop further scenarios that are more carefully designed for multiple players performing collectively.

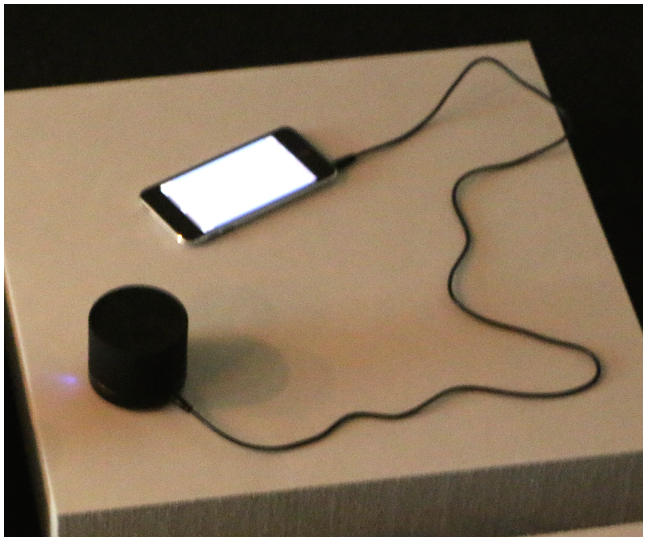

Figure 2: iPod Touch and loudspeaker used by the teenagers in *Soundworks*-based scenarios tested at the *Collective Sound Checks*.

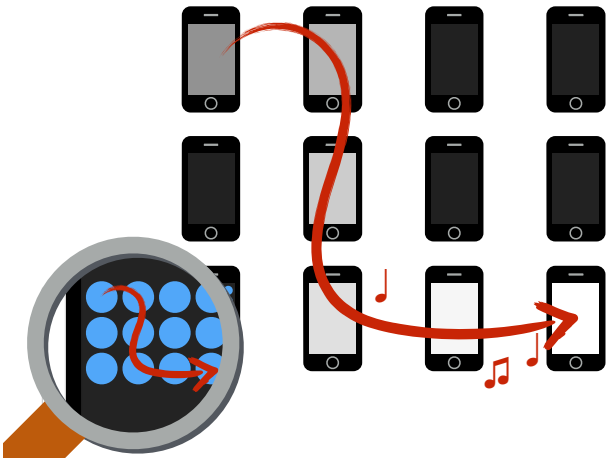

Figure 3: *Wandering Sound*. The screen of a performer displays the spatial representation of all the smartphones. By moving his / her finger on the screen, this player makes sound and light wander across the matrix of smartphones.

3.2 Drops

The *Drops* performance is strongly inspired by the mobile application *Bloom* [1] and reproduces several audiovisual elements of the original application while transposing them into a collective experience and introducing some new ideas. The players trigger resonant sounds by tapping the screen of their smartphones. Unlike the *Bloom* application, each player can play a single sound (*i.e.* a single pitch) that is varied in timbre depending on the touch position which affords to construct sound sequences (*i.e.* melodies) together with the other players. The sound sequences are repeated every few seconds in a loop, each time more attenuated than the previous one, until they ultimately vanish. The sounds triggered by one player are automatically echoed by the smartphones of two other players, where they are integrated in the loop of repeated and slowly vanishing sounds. Very similar to the *Bloom* application, each sound is visualized by a circle growing from the tapping position and fading with the sound. The players can clear the loop — and the screen — by shaking their smartphones. The collective performance on the smartphones is accompanied by a synchronized soundscape on ambient loudspeakers. The application is illustrated in Figure 4.

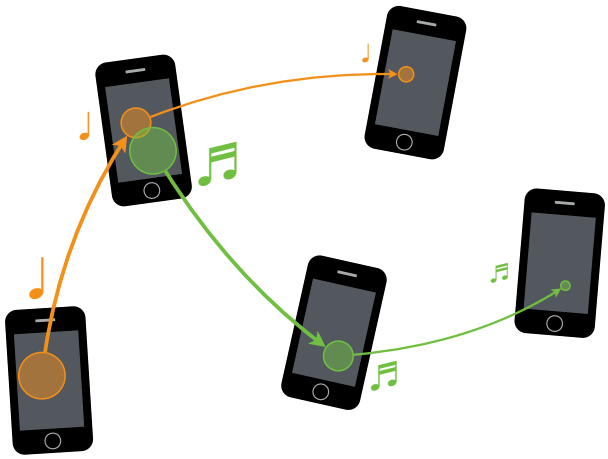

Figure 4: *Drops*. The sound and graphics produced when a player touches the screen are echoed on the smartphones of other players. They are looped a few times before fading out completely.

In our first tests of the application with the teenagers, we observed that it can be challenging for the participants to articulate their performance with the performance of the others players. The teenagers first compensated the single note limitation by playing on the sound density — thus testing the limits of the application — before learning to listen to the others and to collaboratively construct sequences. In further tests we will explore several variants of the application that would either allow for playing multiple sounds on each smartphone or further constrain the possibilities of each player (*e.g.* a limited number of triggers within a given time frame).

3.3 Paths

The *Paths* performance reproduces the basic concept and audiovisual elements of the *Luminaria* environment from the *Electroplankton* game [3]. In this scenario, the players are placed on a matrix grid (for instance, 3 rows and 4 columns). A sound source travels the matrix of smartphones at a steady tempo, following the path determined by the orientation of the smartphones. For instance, when the source arrives on a player's smartphone that points at the player on the right, the source will travel to that player on the next beat and continue its path from there (cf. illustration on Figure 5). Each sound source has the same timbre on all the smartphones but a different note: we use a pentatonic scale so that the source plays a melody as it travels the matrix of smartphones. We can also add multiple sound sources (that have a different sound and use a different rhythm such as 8^{th}-notes or triplets) in order to enrich the performance.

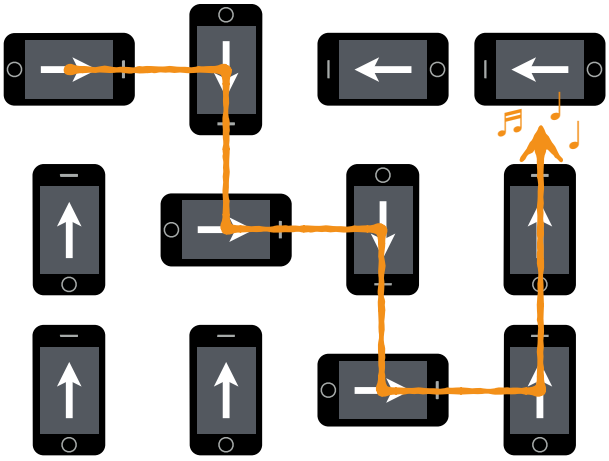

Figure 5: *Paths*. The players can orient their smartphones in order to direct a sound source that travels the grid of smartphones at a steady tempo.

Testing the application with the teenagers made us realize a number of traps to avoid in the design of collaborative performance scenarios. For example, while many performance scenarios may look interesting when looking at all the smartphones from above (*i.e.* laid out on a tabletop during the development), the experience of the performers interacting with each other is generally very different if they can't see the whole setting while playing. For *Paths*, this suggests developments where a visualization (screen, mirrors on the ceiling) would reflect a global view of the performance.

4. CONCLUSION AND FUTURE WORK

With *Soundworks*, we intend to contribute a flexible framework for creating collaborative performance scenarios based on web technologies. In these scenarios, the participants are both performers and listeners engaging in audiovisual interactions supported by a network infrastructure. Ultimately, *Soundworks* provides an ubiquitous and configurable setup to experiment with novel ways of music-making.

By making the framework modular, we hope that artists and developers will find it easier to create their own scenarios, and expect the community to grow and improve the modules library. The boilerplate and modules are available in the `Ircam-RnD` organization[2] on GitHub, under BSD-3-Clause license, as the `soundworks` repository [7]. The repository includes documentation and working examples.

Testing the three *Soundworks*-based scenarios with groups of teenagers at the Centre Pompidou allowed us not only to validate the concepts and implementation of this framework, but also to identify compelling future developments.

Apart from creating further scenarios as well as additional modules that artists and developers may reuse, the most important challenge we have to solve is the disparities across smartphone capabilities. For instance, we noticed that some lower-end smartphones did not have the necessary networking and/or computing capacities to run the scenarios smoothly. Other important inconsistencies across different smartphone models concern the audio. For instance, we noticed that the audio output latency can vary by hundreds of milliseconds across different models, which makes sound spatialization and beat synchronization much more challenging. An additional obstacle, especially to create sound spatialization effects, is the uneven output level of the speakers.

Since we want anyone to be able to participate in *Soundworks*-based scenarios spontaneously with his or her own smartphone, an immediate next step of the development will be to ensure that a wide range of scenarios work on a wide range of different smartphones. This includes the revision of the sensor input modules to provide alternative implementations of the interactions if certain sensors are not available, as well as the compensation for level differences and audio latencies to allow for beat synchronizing an inhomogeneous set of smartphones. In addition, future versions of the framework will support the implementation of alternate versions of the scenarios, which may ultimately allow to attribute different — musical — roles to the players depending on the capacities of their smartphone.

[2]https://github.com/Ircam-RnD

5. ACKNOWLEDGEMENTS

The developments described in this article were conducted in the context of the *CoSiMa* research project, partially funded by the french National Research Agency (ANR, project ANR-13-CORD-0010). We would like to thank our colleagues Frédéric Bevilacqua, Diemo Schwarz, Victor Saiz, Benjamin Matuszewski, Jean-Philippe Lambert and Lorenzo Bianchi, as well as our partners of the Studio 13/16, Orbe, and NoDesign for their valuable contributions to this work.

6. REFERENCES

[1] B. Eno and P. Chilvers. Bloom. `http://www.generativemusic.com/bloom.html`, 2012.

[2] G. Essl and M. Rohs. ShaMus – A Sensor-Based Integrated Mobile Phone Instrument. ICMC'07, pages 27–31, Copenhagen, 2007.

[3] T. Iwai. Electroplankton. `http://www.nintendo.com/consumer/gameslist/manuals/DS_Electroplankton.pdf`, 2005.

[4] S. W. Lee and J. Freeman. *echobo*: A Mobile Music Instrument Designed of Audience to Play. NIME'13, Seoul, 2013.

[5] J. Oh, J. Herrera, N. Bryan, L. Dahl, and G. Wang. Evolving The Mobile Phone Orchestra. NIME'10, Sydney, 2010.

[6] J. Reid, R. Hull, T. Melamed, and D. Speakman. Schminky: The Design of a Café-Based Digital Experience. *Personal Ubiquitous Computing*, 7(3-4):197–202, July 2003.

[7] S. Robaszkiewicz and N. Schnell. Soundworks boilerplate. `https://github.com/Ircam-RnD/soundworks`, 2014.

[8] A. Tanaka. Mobile music making. NIME'04, pages 154–156, Singapore, 2004.

[9] D. Trueman, P. Cook, S. Smallwood, and G. Wang. PLOrk: The Princeton Laptop Orchestra, Year 1. ICMC'06, Aug. 2006.

[10] G. Wang. Designing Smule's iPhone Ocarina. NIME'09, Pittsburgh, 2009.

[11] G. Wang, G. Essl, and H. Penttinen. MoPho: Do Mobile Phones Dream of Electric Orchestras. ICMC'08, Belfast, 2008.