massMobile – an Audience Participation Framework

Nathan Weitzner
Georgia Tech Center for Music Technology
840 McMillan St.
Atlanta GA 30332-0456
nweitzner3@gatech.edu

Jason Freeman
Georgia Tech Center for Music Technology
840 McMillan St.
Atlanta GA 30332-0456
jason.freeman@gatech.edu

Stephen Garrett
School of Interactive Computing
Georgia Tech
85 Fifth Street NW
Atlanta GA 30332-0760
stephen.garrett@cc.gatech.edu

Yan-Ling Chen
Georgia Tech Center for Music Technology
840 McMillan St.
Atlanta GA 30332-0456
ychen450@gatech.edu

ABSTRACT
massMobile is a client-server system for mass audience participation in live performances using smartphones. It was designed to flexibly adapt to a variety of participatory performance needs and to a variety of performance venues. It allows for real time bi-directional communication between performers and audiences utilizing existing wireless 3G, 4G, or WiFi networks. In this paper, we discuss the goals, design, and implementation of the framework, and we describe several projects realized with massMobile.

Keywords
audience participation, network music, smartphone, performance, mobile

1. INTRODUCTION
As the capabilities of smartphones grow and they become increasingly ubiquitous, they provide an opportunity to facilitate interactive, expressive, and unique collaborative experiences in live musical performances. massMobile is a flexible, scalable framework designed to facilitate large-scale audience participation in performances using smartphones as the communication link between audience members and performers.

Our broad motivation is to enable scalable, creative group collaboration. Despite advancements in technology that facilitate digital networked communication, it remains challenging to facilitate musically engaging collaborative creativity in contexts that require minimal musical or technical proficiency and no extended rehearsal. And as the number of participants grows, it becomes increasingly challenging to maintain a balance between the transparency of individual contributions and the coherency of the collective product. massMobile is not in itself a solution to these challenges, but rather a framework designed to enable rapid, iterative design, development, deployment, and evaluation of systems that explore these challenges.

In this paper, we will first address previous work in audience participation in live performance, followed by massMobile's design goals and implementation. Finally, we describe its use in a series of projects and performances.

2. BACKGROUND
Music has traditionally held a clear distinction between the roles of the composer, performer, and listener. When the audience becomes involved in the performance these distinctions become less defined. Until recently, large audiences have been restricted in the ways they can participate in a performance on account of both logistical and cultural constraints. However, the growing capabilities of digital technology and the pervasive cultural role of social media now provide the opportunity and context for a deeper interaction.

2.1 Participation Devices
There are several approaches currently used to facilitate audience participation, such as tracking the audience's movement and distributing devices to the audience. Maynes-Aminzade describes a system that tracks the motion of the audience using computer vision [10]. Other projects have distributed devices such as light sticks in *Glimmer* [8], lighted hats in *Flock* [7], or colored paddles used in *Cinematrix* [3], which can be tracked using computer vision. Feldmeier and Paradiso hand out disposable wireless sensors to participants [6]. Distributing devices, though, requires expensive investments in equipment and infrastructure as well as significant setup and calibration time at each performance venue. This ultimately limits the feasibility of potential performances. Further, the interface design of each system is typically highly constrained by the hardware, limiting both the depth of audience interaction and the adaptability of the systems to diverse interaction designs.

With the ever-growing ubiquity and capability of mobile technology, artists and researchers have begun to explore audience interaction through these means. Golan Levin's *DialTones* [9] utilizes the audience's mobile phones as a distributed array of loudspeakers. More recent projects use smartphones as platforms for more active participation. In *TweetDreams* [5], audience members tweet from their phones to influence sound and visuals in performance. In *Moori* [11], participants respond to prompts from the performer by typing short messages into a locally-networked application.

Oh and Wang discuss similar smartphone projects and evaluations in the context of audience participation [13].

2.2 Networked Smartphone Interfaces for Music
Two main approaches to networking mobile devices in a performance setting are over local WiFi networks temporarily setup for a performance, and over Internet-based networks.

Local networks typically provide a faster transmission of data and are well suited for applications that demand minimal latency. *TouchOSC* [14], *Mrmr* [12], and *Control* [4] are examples of frameworks that utilize local networks, with the original design intent of replacing dedicated hardware fader boxes for controlling music software with highly-configurable wireless smartphone interfaces.These frameworks are typically intended for single user interaction with a single computer but can support several users. Jesse Allison's Nexus Project [1]

NIME'12, May 21-23, 2012, University of Michigan, Ann Arbor.

works in a similar manner but is deployed as a browser-based interface rather than a native smartphone application in order to leverage web standards and facilitate multi-platform use.

In the context of massMobile, these systems identify fundamental design questions surrounding wireless networks and web standards. While these applications must minimize latency (at the expense of easy network configuration), massMobile must maximize ease of setup (since audience members must quickly install and configure the framework themselves). The use cases for massMobile, in turn, must assume a high latency and incorporate this limitation into their designs. massMobile is also directly influenced by the ways in which these systems use standard tools and reusable, configurable components to facilitate rapid development of interfaces for new contexts.

3. GOALS

massMobile seeks to extend the tradition of audience participation in live performance and explore new possibilities, with a focus on three overarching goals: enabling rapid development of new participatory interfaces; enabling plug-and-play deployment at performance venues; and facilitating simple use by audience members on a variety of smartphone platforms.

Rapid development enables iterative artistic experimentation through the design – test cycle. Iterations of the interaction design can be developed and deployed entirely in software code and/or configuration changes, so they take days (or even seconds) instead of months as with hardware-based systems.

Plug-and-play setup is critical for deployment in a variety of venues where there may not be technical infrastructure to support large open WiFi networks, and there may not be time or access to setup and calibrate hardware such as computer vision systems.

Audience members need the ability to instantly launch an application and participate, ideally in a single step. It is not engaging to dedicate a significant amount of performance time to setup and configuration, nor practical to send assistants through a large audience to help them connect.

4. DESIGN AND IMPLEMENTATION

massMobile consists of a web-client for use by the audience on any recent smartphone connected to any Internet connection (WiFi, 3G, 4G), a Max/MSP API for use in software for the performance, and an offsite server that passes information between the smartphone and Max/MSP clients (Figure 1).

4.1 Smartphone Client

4.1.1 Smartphone client architecture

We initially developed native applications for iOS and Android but eventually abandoned these in favor of a web application. The additional overhead of developing and maintaining two separate applications was contradictory to our design goal of rapid development in extending the framework, and it lacked support for platforms such as Windows Phone 7.

The current smartphone web client uses common web standards including the Google Web Toolkit, HTML5, and SOAP and runs without modification on all major smartphone web browsers. While this development approach lacks access to some native phone APIs, it met the needs of massMobile and performs at near-native speeds. Revisions do not require deployments to native app stores, and audiences need not download a native application. (The web site link is distributed to the audience as both a QR code and a short URL.)

Because massMobile connects to a persistent remote server, it requires no special network configuration. Once audience members load the URL, they are ready to participate.

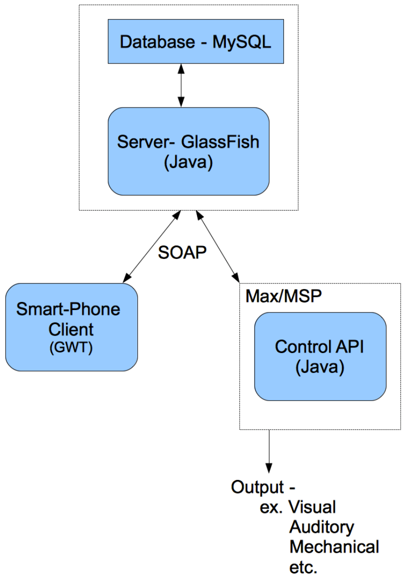

Figure 1: Block diagram of the massMobile framework.

4.1.2 Web Client User Interface

The user interface has a customizable GUI that can be configured for different uses via a server-side database and the Max/MSP API (see below). When a user loads massMobile, the URL's query string automatically loads the configuration for a particular project and event. In massMobile, projects represent user interfaces that may be used across multiple performances, while events represent a single performance of a project and all of its associated audience participation data.

The main page for an event has tabs for introductory text and for each active user interface element (called a "model" in massMobile). Models support interaction modes such as sliders, text entry, voting, and drawing. Parameters of their display are preconfigured through a server database and can also be updated in real time during an event.

massMobile has two ways to provide feedback to audience participants: via a video projection in the performance venue that is generated by the Max/MSP software, or via visual elements that are displayed directly within the smartphone interface. The latter is particularly useful in venues that lack video projection. Feedback typically shows a visualization of audience response (e.g. the tallies of votes for different choices) so that participants can consider the collective activity as they use the interface, and so that they can better follow the influence of audience participation on the live performance.

4.2 Max/MSP API

Because massMobile can be used in a variety of performance contexts, we provide a generalized API for Max/MSP rather than a single, monolithic application to drive performances. Our goal is to enable any application developed with the Max/MSP API to be plug-and-play, so that even performers with minimal technical expertise can launch an application on their laptop and perform with massMobile without assistance or further technical infrastructure (though some massMobile projects inevitably require additional resources for things such as electroacoustic sound diffusion).

The Max/MSP API connects to our server to get and process both raw and pre-processed data, update GUI configurations on connected phones, and simulate audience members' interactions with the system for rehearsal and testing purposes. The API is implemented in Java as an MXJ external and provides a highly-efficient multi-threaded architecture for repeated queries of the server and processing of the response before passing the data to Max/MSP. The API also has methods for searching data based on SQL query strings so that sets of data with specific attributes can be returned.

4.3 Controlling the Smartphone Client from the Max/MSP API

In most massMobile projects, it is necessary to change the smartphone client's GUI configuration in real time during an event. Changes might enable or disable all interaction, step through different interaction interface models, or change text instructions or configuration of a single interface model.

These changes are typically triggered from the Max/MSP API and can be done in two ways. The first is through a pre-programed queue of changes that can be stepped through by pressing the spacebar, using an external controller, or configuring an automatic timer. The framework also allows for spontaneous manual changes of the smartphone interface through a "Configuration Editor" (Figure 2), permitting performers to respond to an event's unique dynamic in real time and to tweak the smartphone interface accordingly.

4.4 massMoblie Server

Our server includes a servlet layer (implemented with Java, SOAP, and GlassFish) that handles requests from the clients, and a database (MySQL) that stores all audience and configuration data. By hosting the server offsite from performance venues, we enable smartphone access over any Internet connection at the expense of latency. By using web standards and a database, we are able to host a robust, scalable server and also to archive participation data for later analysis.

5. MASSMOBILE PROJECTS

We first used massMobile in collaboration with choreographer Jonah Bokaer, where we carried out initial testing of the framework in spring 2011. massMobile was used in his work *FILTER* in performances in Avignon, France and at the Ferst Center for the Arts at Georgia Tech. In addition to these public performances, we conducted several informal trials with small audiences and a solo dancer onstage as we developed the software.

In the initial tests, audience members voted on lighting preferences. The votes were displayed on a video projection and also used to change the lighting configurations, which in turn cued the dancer to change his choreography. The small groups enjoyed responding to each other's votes and seeing the resulting change in dance and lighting, and a group dynamic quickly developed in which "leaders" would change their votes and then "followers" would gradually shift over to one leader, collectively effecting a change. In the public performances, we lacked access to video projection and had not yet implemented feedback on the phones themselves, so we shifted to a simpler interaction paradigm in which audience members triggered the lighting to highlight different scenery elements. The lights flashed on after a button was pressed and then faded out over a few seconds. This approach was less successful; the lack of clear visualization of audience participation hindered the development of a cohesive group dynamic in performance.

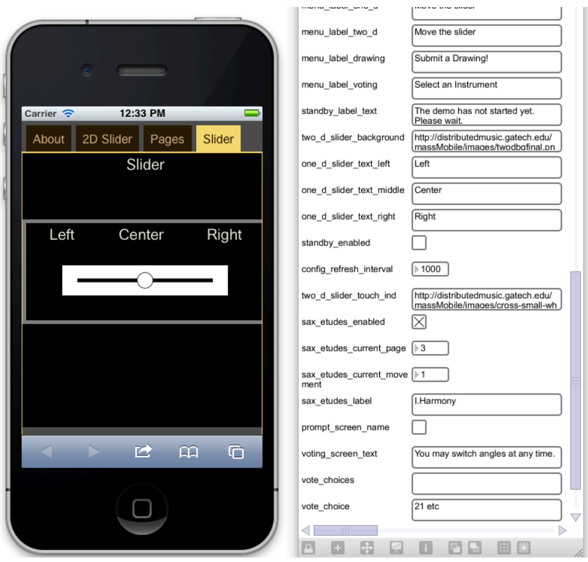

Figure 2: Left: User smartphone interface showing the slider model. Right: Configuration editor in Max/MSP API.

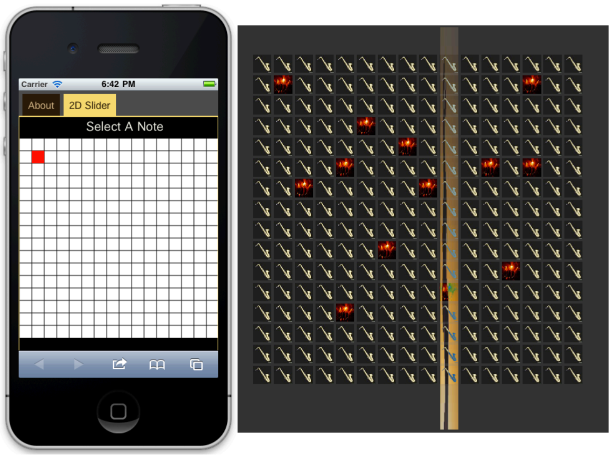

Figure 3: Tone matrix model on the left and the projected tone matrix on the right.

During the fall of 2011, we presented massMobile in a series of demos at Georgia Tech. During these demonstrations participants created music collaboratively through a tonematrix interface, selecting points on a grid to activate specific pitches at specific times (Figure 3). massMobile turned this classic sequencer paradigm into a collaborative improvisation as participants composed a constantly-changing loop together. There were no performers aside from the co-located participants using their mobile devices to interact with the shared tonematrix. Everyone could see the looping sequence projected on a screen and hear the result through a speaker

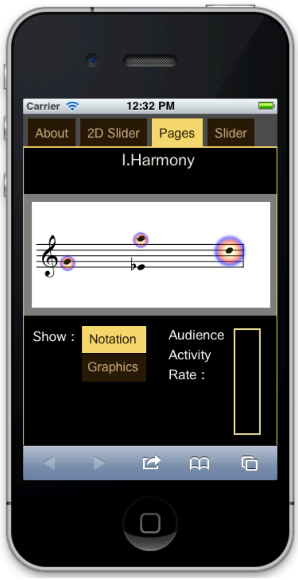

Figure 4: First movement of Saxophone Etudes

system connected to the Max/MSP computer.

We are currently preparing for a premiere this spring of one author's (Freeman) new composition, *Saxophone Etudes*. In this piece the audience votes on which part of the score the saxophonist should play at any given moment by clicking on one of several notes or selecting a portion of a longer musical motive (Figure 4). The distribution of votes is displayed to the performer running the Max/MSP software, superimposed on the notation itself, so that she can use the information to guide her performance of the music. The distribution is also displayed to audience participants on their phones. The saxophonist steps through preset cues to turn the "pages" of the music on her own display and on the smartphones, using either an automatic timer or a foot pedal.

Because latency over cellular networks is both potentially large and extremely unpredictable (and in our experiments can be as long as a second), these massMobile projects have all been designed to accommodate such latencies. Data sent to the server is time-stamped when inserted into the database allowing for reconstruction of the event sequence as received. In the tonematrix demonstration, Max/MSP handles absolute timing while the smartphones place elements on the sequencer grid; such strategies have long been common to address latency in networked music collaborations such as WebDrum [2]. For *Saxophone Etudes*, timing delays have little effect on the overall voting tallies or their visualization.

6. FUTURE DEVELOPMENT

We are currently collaborating with large ensembles and individual artists to bring massMobile to new performance contexts, and we also plan to conduct user studies to help us improve both the framework and the projects we realize with it.

We are developing a new project with massMobile focused on collaborative drawing during a performance. Participants draw on a shared canvas in response to the music during the performance. Afterwards, animation of the canvas is synced to an audio recording of the performance to create both a "music video" of the audience's collaboration and also to serve as a new graphical score to guide improvisation of a new performance.

We are also planning to use massMobile with Georgia Tech's marching band at a football game halftime show in fall 2012, with thousands of fans using their smartphones to influence both the music and formations of 350 musicians on the field.

We plan to conduct the user studies via web-based surveys available from within the massMobile application. This feedback, coupled with quantitative analysis of the database event log data, will provide needed material for formal evaluations of both specific implementations of the system and the overall framework. Additionally, this will provide insight into how audience members perceived their role in the performance, transparency of the interaction, and what they thought about the musical result.

7. CONCLUSION

Rather than being a specific implementation of an audience participation piece, massMobile is a configurable, flexible and scalable framework for collecting and processing audience input. Mappings, processing algorithms, audience feedback, and musical goals are not hard-coded into the framework but are intended to be quickly and easily configured for a specific implementation. With this framework, novel mapping strategies can be quickly tested, algorithms can be modified in real-time to adjust for audience size or amount of activity, and user interfaces can be rapidly developed in order to produce a more collaborative, expressive, and creative musical experience.

8. ACKNOWLEDGMENTS

More information about massMobile is available at http://www.gtcmt.gatech.edu/research-projects/massmobile. In addition to the authors, Sriram Viswanathan contributed towards early development of massMobile. massMobile is supported in part by a grant from the National Science Foundation as part of a larger research project on musical improvisation in performance and education (NSF CreativeIT 0855758).

9. REFERENCES

[1] Allison, J., Distributed Performances Systems using HTML5 and Rails. In *Proceedings of 26th Annual Conference of the Society for Electro-Acoustic Music in the United States (SEAMUS)*, Miami, 2011.

[2] Burk, P. Jammin' on the Web - a new Client/Server Architecture for Multi-User Musical Performance. Proceedings of the 2000 International Computer Music Conference (ICMC 2000), Berlin, Germany, 2000.

[3] Carpenter, L. Cinematrix, Video Imaging Method and Apparatus for Audience Participation. US Patents #5210604 (1993) and #5365266 (1994).

[4] Control. (2009). Retrieved January 10, 2012 from http://charlie-roberts.com/Control/.

[5] Dahl, L., Herrera, J., and Wilkerson, C. TweetDreams: Making Music with the Audience and the World Using Real-Time Twitter Data. In *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*. Oslo, 2011.

[6] Feldmeier, M., and Paradiso, J. A. An interactive music environment for large groups with giveaway wireless motion sensors. *Computer Music Journal*, 31(1):50– 67, Spring 2007.

[7] Freeman, J., and Godfrey, M., Technology, Real-time Notation, and Audience Participation in Flock. In *Proceedings of the Conference of Computer Music*, Belfast, UK, 2008.

[8] Freeman, J., 2008a. Glimmer: Creating New Connections. In *Transdisciplinary Digital Art: Sound, Vision and the New Screen*. Berlin: Springer, 270-283.

[9] Levin, G, et al. *Dialtones*. Retrieved January 10, 2012, from http://www.telesymphony.com/.

[10] Maynes-Aminzade, D., Pausch, R., and Seitz, S. Techniques for Interactive Audience Participation. In *Proceedings of the IEEE International Conference on Multimodal Interfaces*, 14-16 October 2002 Pittsburgh, 15-20.

[11] Moori. (2011). Retrieved January 18, 2012, from http://www.bubblyfish.com/thesis/.

[12] Mrmr. Retrieved January 10, 2012, from http://mrmr.noisepages.com/.

[13] Oh, J., Wang, G. Audience-Participation Techniques Based on Social Mobile Computing. In *International Computer Music Conference*, ICMC, 2011, Huddersfield, Kirklees UK, 2011.

[14] TouchOSC, Retrieved January 18, 2012, from http://hexler.net/software/touchosc.