MMM Ball: Showcasing the Massive Mobile Multiuser Framework

Tim Weißker, Andreas Berst,
Johannes Hartmann, Florian Echtler
Mobile Media Group
Bauhaus-Universität Weimar, Germany
`firstname.lastname@uni-weimar.de`

Figure 1: A massive multiplayer soccer game was implemented on the basis of M^3 framework. 17 users play concurrently on a large public display using their mobile devices.

CHI'16 Extended Abstracts, May 07-12, 2016, San Jose, CA, USA.
ACM 978-1-4503-4082-3/16/05.
http://dx.doi.org/10.1145/2851581.2890232

Abstract
We present the Massive Mobile Multiuser (M^3) framework, a platform designed to enable setup-free, real-time, concurrent interaction with shared public displays through large numbers of mobile devices. Simultaneous interaction of multiple persons with public displays requires either dedicated tracking hardware to detect gestures, or a way for users to interact through their personal mobile devices. The latter option provides more flexibility, but also presents a heightened entry barrier as it often requires installation of custom software.

To address these issues, M^3 enables immediate interaction through the mobile browser without requiring prior setup on the user side, and real-time interaction suitable for fast multiplayer games. We present a live demonstration of our framework at the example of a public game which has already been shown to support up to 17 concurrent users. Despite a resource-constrained environment and an unpredictable selection of client devices, M^3 consistently delivers performance suitable for real-time interaction.

ACM Classification Keywords
H.5.2. [User Interfaces]: Input Devices and Strategies; H.5.3. [Group and Organization Interfaces]: Web-based Interfaces

Author Keywords
public display; mobile device; multi-user interface; real-time interaction

Introduction and Motivation
Although public displays have become a common sight in recent years for applications such as advertisements and entertainment, these displays are still mostly passive information sources. Various approaches exist to enable interaction with such devices, in particular focusing on touch and gestures as interface modalities. However, if a more complex interface with multi-user support is desired, particularly for large groups of 10 or more users, an alternative is to employ the users' personal mobile devices. Unfortunately, this approach suffers from a heightened entry barrier as users are often required to install a custom app before interaction is possible. Given that only a short time window is available to engage potential users in a walk-up-and-use scenario [6], the end result is often that no interaction will take place after all.

To address these issues, we present the Massive Mobile Multiuser (M^3) framework, which allows setup-free, real-time interaction with public displays by utilizing the mobile browser. Depending on the scenario, users only have to visit a web page, which may be presented via a short printed URL or a QR code to start interacting. In particular, our system has a sufficiently low latency between user action and visible reaction of approximately 65 ms to enable interaction even with fast-paced multiplayer games.

In this paper, we give an overview of the architecture of M^3 and present preliminary findings from two real-world deployments in a resource-constrained environment with 97 distinct users (up to 17 concurrently). Although a large body of research on this topic exists, our approach is - to the best of our knowledge - the first to combine setup-free interaction, real-time capabilities and support for a large number of concurrent users. The API as well as all source code for the framework and the sample game are available under an open-source license at https://github.com/mmbuw/massive-mobile-multiplayer.

Related Work
Interaction with public displays for multiple simultaneous users is a topic which has already been explored by numerous researchers. We identify three main directions of research pertinent to our approach: vision-based and browser-based interaction using personal mobile devices, and gestural interaction using sensors embedded into the display.

Vision-based interaction usually employs an approach where the public display is viewed through a live video feed on the mobile device and interaction is done through touch on the video representation. A seminal example is Touch Projector by Boring et al. [1]. Common limitations of this approach are that it requires a custom app to be installed before interaction can take place, and that the mobile device has to be held in a camera-like pose to enable interaction.

Browser-based interaction, on the other hand, emphasizes the aspect of requiring no setup prior to interaction, which is performed through the pre-installed web browser on the mobile device. For example, Kubitza et al. present VEII [5], which allows on-site modificiation of public displays in museums through a mobile device. Geel et al. show PresiShare [4], a setup-less web application for mobile devices, which uses QR codes to share media on multiple public displays. Dingler et al. present uCanvas [2], a web-based framework to employ the mobile device's accelerometer for interaction with the public display.

Both the smartphone-camera- and browser-based research directions do not seem to focus on simultaneous real-time interaction from multiple users. Although both approaches should in theory be capable of supporting multiple concurrent users, this has not been investigated extensively. When multi-user capability is desired, most research currently centers around gesture-based interaction using sensors such as the Kinect. One current example is ShadowTouch by Elhart et al. [3], in which the users' silhouettes are overlaid over the display and augmented with individual selection menus.

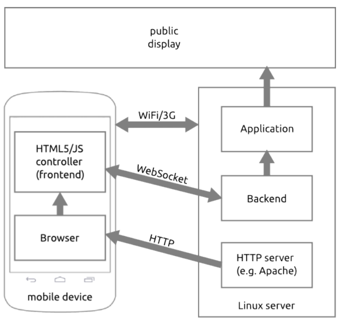

Figure 2: Graphical illustration of M³ framework's architecture. The frontend module runs on the mobile client device; the backend and the application module are deployed on the server to which the public display is connected. To send messages to the the server, a wireless communication channel is used.

Architecture of M³ Framework

The M³ Framework consists of a *client-server architecture*, in which a server performs the application logic and renders a visualization of its internal model to the shared public display. The mobile client devices collect inputs from their respective users for manipulating this model and forward them to the server using a wireless communication channel. While the *frontend* runs on the client devices, the *backend* and the *application* are deployed on the server. The relation between these modules are illustrated in Figure 2.

When a user wants to participate, they wirelessly connect their mobile device to the server. By opening a specific URL in a browser, the connected client is supplied with the *frontend* module of the framework via a regular web server. This module opens a *WebSocket* connection to the backend for realtime communication. In order to collect input data from the user, it displays a virtual input device interface using HTML5 and JavaScript. The frontend currently supports the mobile Firefox, Safari and Chrome browsers. In our demonstration, for example, the frontend presents a virtual game pad with a control stick and a touch button. When an event on the virtual input device is created by the user, it is encoded in a message, sent to the server via the WebSocket connection, and forwarded to the application using standard OS interfaces.

Real-World Deployment of M³

To present our framework in a real-life setting with fast-paced multi-user interaction, we implemented a gaming appliance based on a Raspberry Pi. We assumed that attracting people to test the interface was easier when using a playful application as opposed to an artificial test setup or a more work-focused implementation. Consequently, we developed a video game loosely based off *HaxBall*, a simplified soccer variant. It is well suited as a stress test for our framework as it requires constant interaction and fast reaction times while allowing players to join or leave at any time. This game was deployed during an open-lab event at our university as well as during a digital arts exhibition. For the actual look of the game, refer to Figure 1.

The game constantly repeats rounds of 3 minutes duration, which users can join at any given time. For our system, this is done by simply joining our WiFi network and accessing an arbitrary URL. The user will then be forwarded to the portal page of our game and prompted to choose a user name. The phone then becomes the user's controller (see also figure 3). Users are auto-assigned to the blue or the red team based on current score and team size; team color is also indicated by the UI elements on the frontend. Each user is embodied by a colored player circle on the field.

In a previous public deployment over a total of about 8 hours, we recorded 143 games with user participation. Note that the framework was running continuously and without interruptions for the whole time period. We recorded a total of 97 different MAC addresses and our longest connection recorded lasted for 2106 seconds - 35 minutes. Note that our system disconnects inactive users after 30 seconds, so

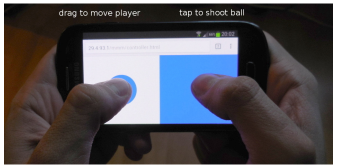

Figure 3: The frontend module of the deployed soccer game using M³ framework obtains relative input values by simulating a joystick; furthermore, a dedicated screen area serves as a discrete button input.

the user had actually been playing for 35 minutes straight. We received enthusiastic informal feedback from participants, and several requests to deploy the game in other contexts such as an office lounge.

To gather additional subjective data, we also redirect users to a short post-game questionnaire in the browser after they have left the game or the 30-second timeout was reached. We ask our users to rate six aspects of their experience: perceived latency, feeling that given input was lost, ease of joining the game, ease of learning to control the player figure, ease of connecting to the system, and expected vs. actual reaction of the figure. User feedback collected from our initial public deployment was very positive.

Conclusion and Future Work

We present the Massive Mobile Multiuser framework, which combines real-time capabilities and setup-free interaction with public displays using personal mobile devices. Results from previous public deployments confirmed our expectations that M³ enables a large number of users to interact concurrently, in real time, and without discouraging setup procedures.

The framework is designed in a modular way in order to provide easy exchangeability of functionalities and use cases. While the source code for the backend module usually remains constant, the frontend and the application are strongly dependent on the framework's usage context. Going from a 2D to a 3D gaming scenario, another imaginable frontend could, for example, provide two control sticks for manipulating both the player's and the camera's movements separately. Apart from gaming, collaborative text editing involving a keyboard frontend might also be an interesting use case for the framework.

REFERENCES

1. Sebastian Boring, Dominikus Baur, Andreas Butz, Sean Gustafson, and Patrick Baudisch. 2010. Touch Projector: Mobile Interaction Through Video. In Proc. CHI '10. ACM, New York, NY, USA, 2287–2296. DOI: http://dx.doi.org/10.1145/1753326.1753671
2. Tilman Dingler, Tobias Bagg, Yves Grau, Niels Henze, and Albrecht Schmidt. uCanvas: A Web Framework for Spontaneous Smartphone Interaction with Ubiquitous Displays. In *Human-Computer Interaction – INTERACT 2015*. Number 9298 in Lecture Notes in Computer Science. Springer International Publishing. http://link.springer.com/chapter/10.1007/978-3-319-22698-9_27
3. Ivan Elhart, Federico Scacchi, Evangelos Niforatos, and Marc Langheinrich. 2015. ShadowTouch: A Multi-user Application Selection Interface for Interactive Public Displays. In Proc. PerDis '15. ACM, New York, NY, USA, 209–216. DOI: http://dx.doi.org/10.1145/2757710.2757735
4. Matthias Geel, Daniel Huguenin, and Moira C. Norrie. 2013. PresiShare: Opportunistic Sharing and Presentation of Content Using Public Displays and QR Codes. In Proc. PerDis '13. ACM, New York, NY, USA, 103–108. DOI: http://dx.doi.org/10.1145/2491568.2491591
5. Thomas Kubitza, Sascha Thullner, and Albrecht Schmidt. 2015. VEII: A Toolkit for Editing Multimedia Content of Interactive Installations On-site. In Proc. PerDis '15. ACM, New York, NY, USA, 249–250. DOI: http://dx.doi.org/10.1145/2757710.2776806
6. Jörg Müller, Florian Alt, Daniel Michelis, and Albrecht Schmidt. 2010. Requirements and Design Space for Interactive Public Displays. In Proc. MM '10. ACM, New York, NY, USA, 1285–1294. DOI: http://dx.doi.org/10.1145/1873951.1874203