

uCanvas: A Web Framework for Spontaneous Smartphone Interaction with Ubiquitous Displays

Tilman Dingler, Tobias Bagg, Yves Grau, Niels Henze, Albrecht Schmidt

► **To cite this version:**

Tilman Dingler, Tobias Bagg, Yves Grau, Niels Henze, Albrecht Schmidt. uCanvas: A Web Framework for Spontaneous Smartphone Interaction with Ubiquitous Displays. 15th Human-Computer Interaction (INTERACT), Sep 2015, Bamberg, Germany. pp.402-409, 10.1007/978-3-319-22698-9_27. hal-01609421

HAL Id: hal-01609421

<https://hal.inria.fr/hal-01609421>

Submitted on 3 Oct 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



uCanvas: A Web Framework for Spontaneous Smartphone Interaction with Ubiquitous Displays

Tilman Dingler¹, Tobias Bagg², Yves Grau³, Niels Henze¹, Albrecht Schmidt¹

VIS, University of Stuttgart, Germany

¹firstname.lastname@vis.uni-stuttgart.de,

²tobiasbagg@googlemail.com, ³grauys@googlemail.com

Abstract. In recent years the presence of displays has become ubiquitous. They range from small-sized screens, such as smartphones or tablets to large screens as they are found in projection screens or public displays. Each display requires a unique modality of interaction, such as a dedicated input device, direct touch or does not provide any interaction at all. With the ubiquity of smartphones people carry with them a high-end interaction device that can connect to any web-connected screen. To allow quick access, we built *uCanvas* ("Ubiquitous Canvas"), a system to engage with interactive surfaces. In contrast to previous work no additional hardware is required, nor do users need to install any proprietary software. Our system runs on all current smartphones equipped with magnetometer and accelerometer, which is used to define a canvas and transmit cursor positions to a server connected to the display. To integrate interactive surfaces into applications, we created a lean Javascript library that allows publishers to specify interaction parameters (such as pointing, clicking, menu selection and text entry) by adding just a few lines of code. We built two example applications to evaluate the feasibility of the system and findings show that (1) interaction is intuitive and (2) easy to set up on the user side.

Keywords: mobile interaction, public display, ubiquitous computing, magnetometer, pointing, smart environments, toolkit.

1 Introduction

Mark Weiser's vision of integrating computers seamlessly into our everyday environment [8] has taken shape in a ubiquity of displays: screens are present almost everywhere in different shapes, sizes, functionality (informative vs. interactive), location and context (home, work, public). Especially public displays have been emerging in recent years being installed in airports, train stations, shopping malls as well as outside. Their primary function still lies in being a sole information display with few cases of options for user interaction. Touch-based interaction in public spaces is often limited because of screen protection (e.g. against vandalism). Hence, these displays tend to provide only unidirectional broadcast without possibilities for bidirectional interaction. In home environments on the other hand, appliances like projectors, TV

or other device screens mostly lack sophisticated input capabilities beyond those of a TV remote. Meanwhile, people carry with them the most ubiquitous personal computing device: their mobile phone. Previous work has taken into account mobile phones as input devices ([1], [3]) for large screens. However, none of these techniques have been widely adopted. Often times the setup is costly and requires additional hardware infrastructure. In other cases the user interaction is cumbersome due to the fact that proprietary software needs to be installed first, which disrupts the flow from user intention to actual interaction.

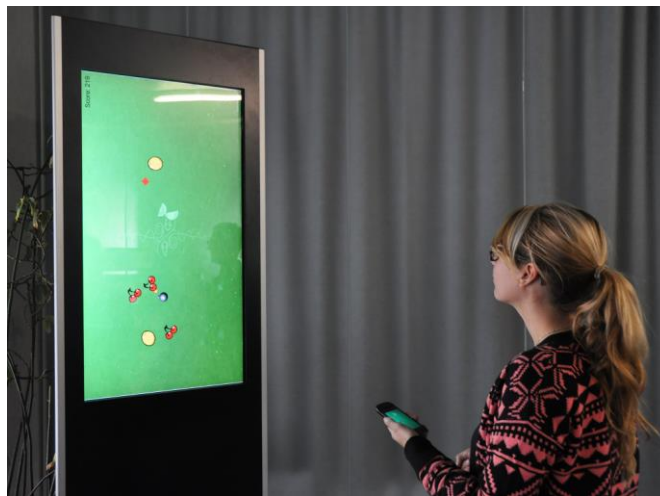


Fig. 1. User interacting through *uCanvas* with the game *food ninja* on a public display.

To provide an easy and quick way to make any display interactive and allow spontaneous interaction for users equipped with a smartphone, we developed *uCanvas*: the system consists of three components: any kind of screen connected to the web, the user's smartphone with integrated accelerometer and magnetometer, and the *uCanvas* server. To interact with the screen, the user follows three simple steps: (1) device to screen registration is done via entering the canvas URL. (2) To calibrate the canvas, the user points the device to the screen's upper left corner, taps and releases at the bottom right corner. (3) Canvas interaction is done by simply pointing the device within the defined corners (Fig. 1). We use the phone's built-in magnetometer and accelerometer sensors to map the user's pointer to a cursor position on the screen. Additional functionality on the smartphone can be provided beyond the continuous cursor positioning: left-click, right-click, a menu or a keyboard for text entry. Multiple users can simultaneously interact with one canvas and the same canvas can be shared and displayed in multiple locations allowing remote interaction.

Screens that can be made interactive in this way include projected spaces, public displays, media facades, TV or smartboards, for example. In fact, any surface showing a website that is connected to the Web, can be turned into a *uCanvas*. For publishers the *uCanvas* system provides an easy way to make their content interactive: we

developed a simple Javascript library, that publishers can include in their website in a few lines of code (Fig. 4 (right)). It is a low-cost solution due to the fact that no new communication protocol needs to be implemented and users can use their own devices without the need to install proprietary software. To assess the feasibility of our approach, we implemented and tested two example applications showing how through minimum effort the requirements of a quick and easy interactive canvas can be addressed, for which pointing and selection works conveniently. The contribution of this paper is 2-fold:

1. We developed a system that allows spontaneous interaction between users' smartphones and any web-connected screen via magnetometer data and
2. that provides a lean Javascript library which publishers can easily integrate to make their content interactive.

2 Related Work

Early work on managing graphic space on large displays using voice and gesture was done in 1980 by Bolt [2]: an entire *Media Room* was designed consisting of a speech recognizer and a complex system of magnetic sensor cubes to recognize pointing direction. Nowadays, there is a great body of research on the interaction with public and distant displays. The most common approach is relative and indirect pointing as well as the use of augmented reality. Ballagas *et al.* [1] make use of a phone's camera to control a mouse pointer on distant displays by using optical flow image processing. Their studies resulted in high task completion time and low scores in the subjective evaluation. Boring *et al.* [3] used the phone's keys, camera and accelerometer to move a pointer across a display. Hardy and Rukzio [6] created *Touch & Interact* where a mobile phone is used to touch a display equipped with a grid of NFC/RFID tags to perform selection tasks. A more novel approach to control appliances is taken by Derthick *et al.* [4]: they use a WiFi access point to display the appliance's web UI as soon as a mobile device connects to it.

There are numerous approaches to implement screen interaction, but there is still no wide-spread adoption. We think this is because costly implementations, cumbersome setup and approaches that are not designed to scale. One way to rapidly deploy a multi-touch enabled display based on projection was suggested by Hardy and Jason [5]. By providing a toolkit they allow designers to deploy interactive surfaces at relatively low costs. Their solution is based on the *Microsoft Kinect*, which allows the detection of touch events on the projected surface. In their case, however, additional hardware is required. Seifert *et al.* [7] on the other hand make use of the phone to point at remote screens. However, the setup is not trivial and cannot be integrated easily by application developers.

Our approach provides a scalable client-server solution where scale is reached by allowing a multitude of devices to be connected with multiple canvases at the same time. *uCanvas* can be deployed on any screen with a web connection and applies to a wide range of smartphones independent from the phone's platform and solely requiring access to magnetometer readings. Magnusson *et al.* [6] use the phone's magne-

tometer combined with GPS data for non-visual orientation and navigation tasks by allowing users to scan their environment by pointing. With pointing being a natural gesture we used this approach to map users' wrist movements to continuous cursor movements on the screen. In contrast to previous work our approach is optimized for setup speed and without the need for additional hardware infrastructure.

3 System Overview

uCanvas is a multi-user client-server system (Fig. 2). We distinguish between two types of clients: smartphones, which send sensor data and information about the actions to be executed (user clients), and screens connected to the server through a web browser (canvas). The *uCanvas* server routes actions from the smartphone to the connected display where the data is processed by a Javascript library. Each canvas' state is stored on the server, so a canvas can be connected at any time and retrieve its current state. Multiple user clients can be connected to one canvas instance on the server. This setup also allows for sharing a canvas across multiple locations.

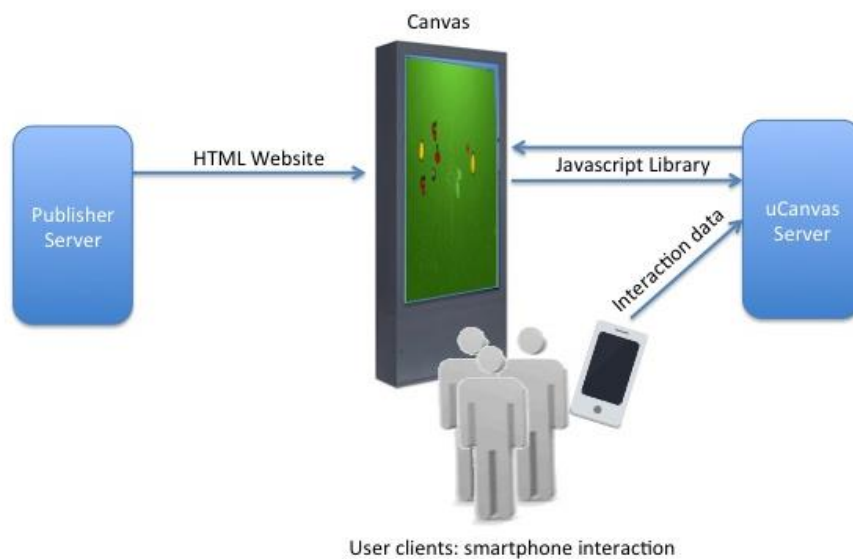


Fig. 2. A Javascript library connects publishers' content with the *uCanvas* server for interaction.

User client. We created two versions of the user client: a native Android app that connects to the *uCanvas* server by opening a TCP connection which is kept alive, and a HTML5 client to allow platform independence and prevent users from having to download a proprietary application. It uses web sockets to keep the connection open to the server. To connect to a screen there are 3 steps: (1) the display registration, where users enter a website URL (Fig. 4 (left)) or scan a code (i.e. barcode, QR code)

to get to the website corresponding to the current canvas which is delivered by the *uCanvas* server. This website contains the canvas-specific parameters which define the possible user actions (pointing, button-clicks, menu and keyboard). (2) The actual canvas needs to be calibrated: The user defines the canvas' bounding box by pointing to the left upper corner of the screen, touching a button and doing the same for the right bottom corner. At button press, the system records the readings from the magnetometer, translates them to relative coordinates and thereby defines a bounding box of the screen. Once calibrated, (3) the user can start interacting by pointing within the bounding box and executing actions on the screen.

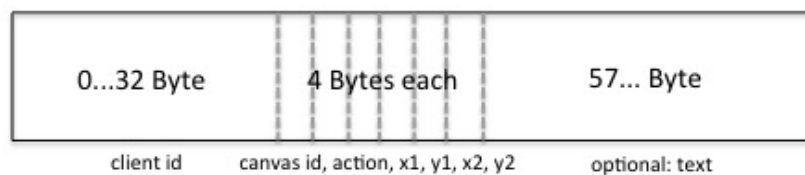


Fig. 3. Data format: Byte array with dedicated positions.

The user faces the screen during calibration and interaction. Because we are using the phone's magnetometer data, user gestures are restricted to wrist movements since lateral arm movements would distort the angle to the screen and therefore the relative position to the bounding box. To keep latency between the phone's movement and the cursor on the canvas at a minimum, the data exchange format consists of a compressed array of 56 Byte as depicted in Fig. 3.

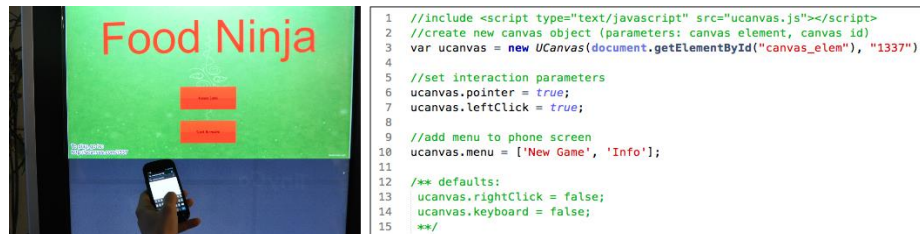


Fig. 4. Left: User connects to *uCanvas* by entering the canvas' URL. Right: Adding interactivity to the UI on the left by adding a few lines of Javascript: defining a pointer, a left-click button and a menu.

Javascript library. This library lets publishers configure parameters for interacting with their website. It provides means to include buttons as well as a menu and keyboard. Hence, content publishers can integrate *uCanvas* into any area of their application by simply declaring which HTML canvas element shall be enhanced with interactivity. Cursor position, clicks as well as text entry are then relayed to the underlying canvas and can be processed. To equip a screen with means of interaction, content publishers follow 3 simple steps: (1) they need to register their website at our

uCanvas server in order to retrieve a canvas id. (2) With this id, a *uCanvas* object can be instantiated on their website via Javascript. Fig. 4 (right) shows how to set up a canvas in 4 lines of code that comes with pointing and click capabilities as well as with a menu. (3) Last, the corresponding canvas URL (or a corresponding barcode or QR code) needs to be displayed somewhere on the screen.

4 User Study

We implemented two example applications to evaluate the feasibility and usability of the *uCanvas* system: a game (*Food Ninja*) and an application for creating mindmaps. We recruited 10 participants (all male), mostly from the academic field, their average age being 25 (SD=3.37). We asked participants to connect to the *uCanvas* (a projection area), calibrate their phone and to familiarize themselves with using the phone for pointing on the display, after which we randomly started with one of the example applications and asked participants to do the following tasks: (1) Mindmap application: first, we presented an existing mindmap and asked them to edit it by deleting nodes and creating new connections. As a second task participants had 4 minutes to create a mindmap from scratch for the topic 'vacation planning'. (2) A game application: We implemented the game *Food Ninja* where players cut fruits that fly across the screen, thereby trying to dodge the occasional bombs. Participants were asked to play 3 rounds, each round ended by contact with a bomb.

Results. We used a System Usability Score (SUS) questionnaire and the general part of the questionnaire for User Interaction Satisfaction (QUIS) to assess the usability of the interactions, followed by a semi-structured interview. Asked about interacting with the display using our system participants provided SUS scores between 67.5 and 95 with an average of 75.75 (SD=10.48). Overall, the system received an average rating of 3.20 on the QUIS general part's 5-point scales (1=negative to 5=positive). Fig. 5 shows the rating for the five individual dimensions.

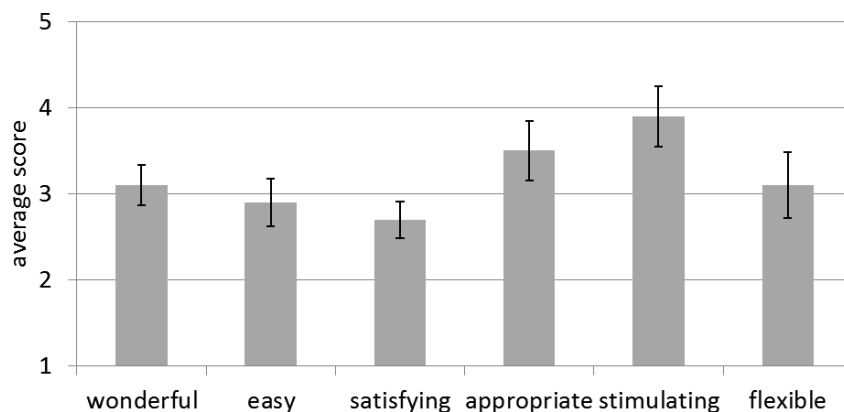


Fig. 5. The average score of the QUIS's general part (1=negative to 5=positive). Error bars show the standard error.

Asked what they liked about the system, participants appreciated its intuitiveness. *E.g.* P9 stated that the system is similar to the *Wii Control* and therefore easy to use. P2 further highlighted that the system is intuitive because of the pointer displayed on the screen. P3 liked that the phone's movement is directly mapped to the pointer's movement and similarly P7 stated that the system enables a very direct input. All participants but P10 and P8 criticized jerky pointer movement. P1 and P3 therefore proposed to smoothen the pointer movement, *e.g.* by averaging the pointer's position over a time window (P1). Other main aspects of criticism have been an observable delay in pointer movement (P2, P5, P8, and P10) and not being able to point with the whole arm but only with the wrist (P1, P6, and P7).

All participants stated that they could imagine to use the system for pointing on a public display. P1, P2, and P9 highlighted that it would particularly useful for gaming. Eight of the ten participants could also imagine using the system at home, particularly for large projected screens and multiple users (P1). Asked about further applications, we received diverse answers. Examples include using the system for presentations (P2 and P10) and games (P1, P3, P6, P7 and P9). Furthermore, they proposed the usage of *uCanvas* for distant displays that cannot be reached through touch (P4 and P9) (*e.g.* like ordering food from a screen at a drive-through) and, in particular, for situations that require quick access (P6 and P9).

5 Discussion

Despite occasional delays and jerkiness of the pointing activity, study participants liked the intuitive interaction and the quick setup. Even though magnetometer and the web connection sometimes behave in unreliable ways, the ease of setting up the system at any web-connected screen and the simple device registration on the user's side make the system being convenient enough for adding interaction capabilities. Since *uCanvas* is based on the phone's magnetometer data, users need to stay in a fixed position once the canvas has been calibrated. However, gyroscope and accelerometer data could be used to detect position change and adjust angles so that users can move around while interacting with the screen. The advantage of our approach is the scalability: users can interact with any size of screen ranging from Desktop screens to large screens across multiple devices. In our user study we collected solely subjective measures. Rigorous system tests including communication delays and pointer accuracy could yield quantitative results. Further, we will need to hand our Javascript library to developers and content publishers in order to collect more detailed feedback on its ease of use and integration.

6 Conclusion

To make any web-connected screen interactive, we developed *uCanvas*, a system that optimizes for easy deployment and quick user setup. Interactions include continuous pointing, clicks, menu selection and text entry. By providing a lean Javascript library we allow publishers to integrate interactive canvases into their applications

with few lines of code. Using the phone's magnetometer and accelerometer allows for natural interaction without the need to install any additional software. We developed two example applications, which we tested with 10 participants to assess the ease and usability of our approach. Future work will focus on automatic canvas calibration by using, for example, highly accurate GPS locations to determine the exact user position in relation to the canvas position nearby. Hence, registration and calibration could be done automatically. Further, we are working on including gesture recognition to extend the capabilities of our Javascript library.

7 Acknowledgements

The authors acknowledge the financial support of the Future and Emerging Technologies (FET) programme within the 7th Framework Programme for Research of the European Commission, under FET grant number 612933 (RECALL). We further thank our study participants.

8 References

1. Ballagas, R., Rohs, M., and Sheridan, J. G. Sweep and point and shoot: phonecam-based interactions for large public displays. In *CHI'05 extended abstracts on Human factors in computing systems*, ACM (2005), 1200–1203.
2. Bolt, R. A. Put-that-there: Voice and gesture at the graphics interface. In *Proceedings of the 7th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '80, ACM (New York, NY, USA, 1980), 262–270.
3. Boring, S., Jurmu, M., and Butz, A. Scroll, tilt or move it: using mobile phones to continuously control pointers on large public displays. In *Proceedings of the 21st Annual Conference of the Australian Computer-Human Interaction Special Interest Group: Design: Open 24/7*, ACM (2009), 161–168.
4. Derthick, K., Scott, J., Villar, N., and Winkler, C. Exploring smartphone-based web user interfaces for appliances. In *Proceedings of the 15th international conference on Human-computer interaction with mobile devices and services*, ACM (2013), 227–236.
5. Hardy, R., and Rukzio, E. Touch & interact: touch-based interaction of mobile phones with displays. In *Proceedings of the 10th international conference on Human computer interaction with mobile devices and services*, ACM (2008), 245–254.
6. Magnusson, C., Molina, M., Rasmus-Grohn, K., and Szymczak, D. Pointing for non-visual orientation and navigation. In *Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries*, ACM (2010), 735–738.
7. Seifert, Julian, Andreas Bayer, and Enrico Rukzio. "PointerPhone: Using mobile phones for direct pointing interactions with remote displays." *Human-Computer Interaction—INTERACT 2013*. Springer Berlin Heidelberg, 2013. 18-35.
8. Weiser, M. The computer for the 21st century. *Scientific american* 265, 3 (1991), 94–104.