System configuration for an audience-driven piece including sonification and visualization of audience's operations

Kita Toshihiro
http://tkita.net

Introduction

Recent penetration of smartphones has made audience participation in live electronics pieces easily possible. The piece played by the author at Csound Conference 2013 is based on the data sent from audience by using their own smartphones or tablets without having to install any dedicated app. The system uses a WiFi router for accepting connection from each participant, and a Linux laptop that runs Csound and lighttpd. By performing XMLHttpRequest data transmissions from the smartphone browsers, OSC message is generated by a PHP script to control Csound.

This paper describes the system configuration for the piece. The revised system that additonally includes processing by HTML5 Web Audio API and Websocket that were added to enhance feedback to each participant is also described.

I. First piece at Csound Conference 2013

The piece "Audience's Smartphone Jam Session" [1,2] that was played at Csound Conference 2013 in Boston is consisted of the data transmitted from audience by using their own smartphones or tablets in the concert venue in real time. What participants have to do is to find the WiFi access point, connect their smartphones to it and go to the Web page where they can specify and transmit x-y data by touching a rectangle area. By performing asynchronous JavaScript data transmissions from the smartphone browsers, OSC (Open Sound Control) messages are generated by a PHP script to control Csound. All the files used or related to this piece including Csound csd files are publicly available at the author's website [3].

As shown in Fig.1, the system for the piece uses a WiFi router for accepting connection from each participant, and a Linux laptop that runs Csound and lighttpd. It uses only the local network and needs no Internet connection.

The people at the concert venue who want to participate in the piece are requested to follow these instructions:

Before the piece begins,
connect your phone or tablet to WiFi '10.0.0.9'
and browse the URL
http://10.0.0.9
to display the touching area.

Touch your phone or tablet during

the first part and the third part.

If a participant completes the instruction, the Web browser of the user displays the rectangle area depicted in Fig. 1. The area is created using HTML5 canvas element (Fig. 2) in order to detect the position where the user touched on.

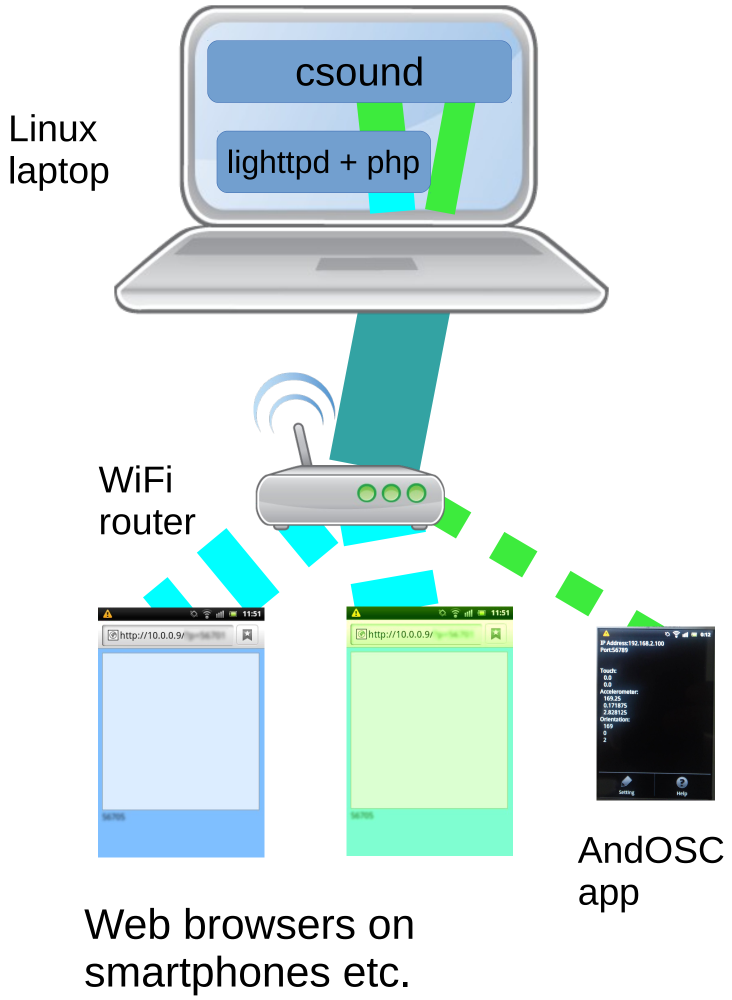

Fig. 1 System configuration for the piece

```
<html lang="ja-JP">

<head>

<meta name="viewport" content="width=device-width,
 initial-scale=1.0, maximum-scale=1.0,
 user-scalable=no" />
<meta charset="UTF-8">
<style type="text/css" media="screen">
 body {  -webkit-text-size-adjust: 100%; }
</style>
<script type="text/javascript">
 var box;
 window.onload = function() {
  box = document.getElementById("touchBox");
  box.addEventListener("touchstart", touchHandler,
   false);
 }
 function touchHandler(e){
  e.preventDefault();
  var xhr = new XMLHttpRequest();
  xhr.open("GET",
   "./oscsend.php?x="+e.touches[0].pageX+
   "&y="+e.touches[0].pageY+
   "&p="+<?php echo $_GET['p'];?>,  true);
  xhr.send();
  var ctx = box.getContext('2d');
  ctx.beginPath();
  ctx.arc(e.touches[0].pageX, e.touches[0].pageY,
   20, 0, 2*Math.PI, true);
  ctx.fillStyle = "rgba(100, 200, 100, 0.2)";
  ctx.fill();
  ctx.lineWidth = 1.0;
  ctx.strokeStyle = "rgba(50, 200, 50, 0.8)";

  ctx.stroke();

 }

</script>

</head>

<body style="background-color:#ff7f7f">

<canvas id="touchBox" style="border:#999999 1px solid;

 background-color:#ffdbdb" width="300px" height="300px">

</canvas>

</body>

</html>
```

Fig.2 HTML and JavaScript source code enabling audience operation on the their smartphones

When a JavaScript event 'touchstart' has occurred, in other words, when the screen of the smartphone of the audience has been touched, touchHandler (defined in the middle of Fig. 2) function is called to execute data transmission to the Linux laptop and drawing of a small circle (centered on the touched point) on the canvas element. Data transmission to the Linux laptop is carried out using 'open' and 'send' methods of the XMLHttpRequest object. Installing a dedicated app is unnecessary as it is totally Web browser operation.

'Width' and 'height' of the canvas element are set to 300px so that the values of x-coordinate and y-coordinate of the touched point are 0 to 300 regardless of the size of the screen of the device. In addition, for stable operation of touching the desired points,

```
<meta name = "viewport" content = "width = device-width, initial-scale = 1.0,
maximum-scale = 1.0, user-scalable = no "/>
```

is set as in Fig. 2 to avoid unintended pinching in and pinching out.

Data from the audience's smartphones, etc., are sent to a PHP script 'oscsend.php' installed on the Ubuntu Linux laptop. For this system, lighttpd [4,5,6] is used to run PHP scripts. The script 'oscsend.php' receives the data and converts the received data using OSC.php [7] to OSC messages that will be sent to Csound.

Although the users do not have to install any apps for participating the piece, the system is designed so that an app called 'andOSC' [8] can also be used for sending data to Csound. The app is able to send OSC messages that include information about touched position on the smartphone screen, acceleration values, and the direction angle of the smartphone.

II. Revised version of the piece at WAC2014

The revised version of the piece was performed at WAC2014 [9] in Paris. The revised piece is accompanied with audio feedback to audience. The system was revised to include sound generation by HTML5 Web Audio API at each participant's smartphone individually. The volume of the smarphone sound is controllable on the Linux laptop. All the smartphones at a time, or selected one or two smartphones can be separately volume-controlled via a Web browser.

For specifying the volume of sound from each smartphone, HTTPD on the Linux laptop return some values to each smartphone as a respond to the HTTP Get request that is sent from each participant by touching the smartphone screen. For generating sound from each smartphone, instances of OscillatorNode is created every time the smartphone screen is touched.

All the files used or related including Csound csd files are available at the author's website [10].

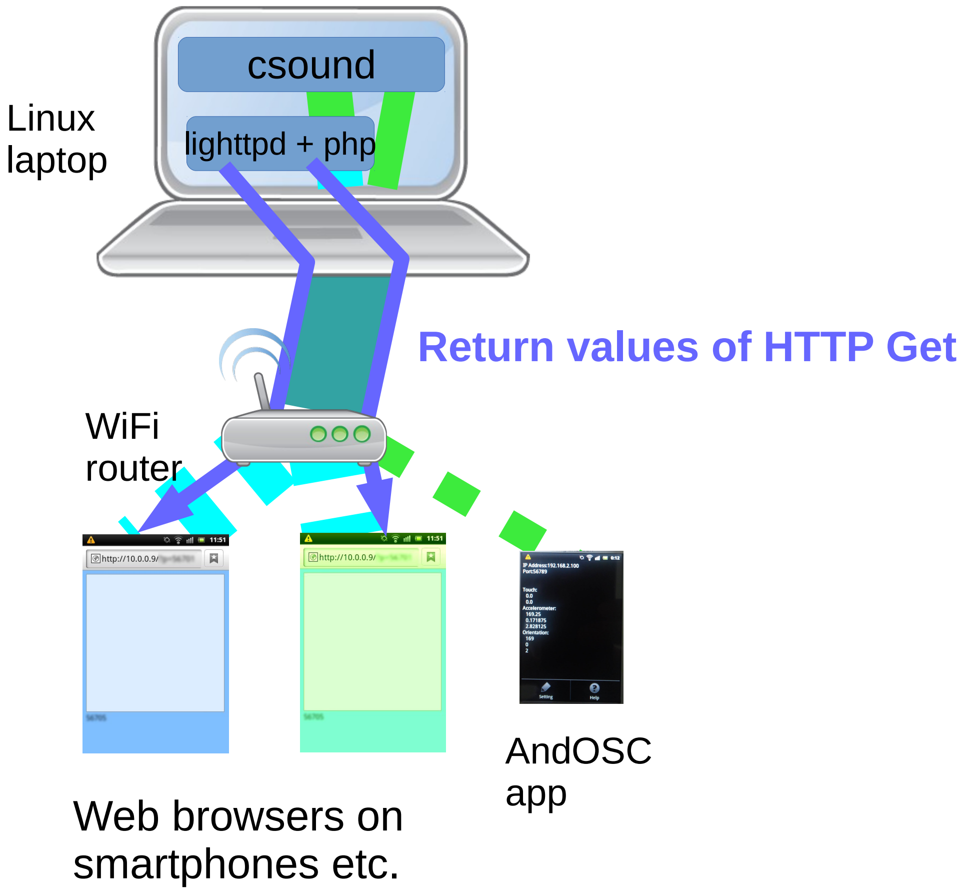

Fig. 2 System configuration for the revised piece

III. Adding visualization of audience operations

To enhance feedback to audience, some visualization of data sent from each participant can be added. Visualized image can be viewed in real time on Web browsers by using Websocket [11] protocol and D3.js [12]. All the related files will be available at the author's website [13].

References

[1] Kita Toshihiro, “Audience's Smartphone Jam Session (excerpt)” at Concert6, 2nd International Csound Conference, Boston, October 27th, 2013.

http://www.youtube.com/watch?v=J12iOcZkI6I

[2] System configuration for ”Audience's Smartphone Jam Session”, an audience-driven interactive piece requiring no dedicated apps, Publication of Japanese Society for Sonic Art, Vol.5 No.3, pp.12–14 (2013.12) (in Japanese) http://data.jssa.info/paper/2013v05n03/3.Kita.pdf

[3] http://tkita.net/csound/conf2013/

[4] Home - Lighttpd http://www.lighttpd.net/

[5] Installing Lighttpd With PHP5 (PHP-FPM) And MySQL Support On Ubuntu 13.04

http://www.howtoforge.com/installing-lighttpd-with-php5-php-fpm-and-mysql-support-on-ubuntu-13.04

[6] lighttpd – Wikipedia http://ja.wikipedia.org/wiki/Lighttpd

[7] Open Sound Control for PHP http://opensoundcontrol.org/implementation/open-sound-control-php

[8] andOSC https://play.google.com/store/apps/details?id=cc.primevision.andosc

[9] Kita Toshihiro, "Smartphone Jam Session with Audience" at Web Audio Gigs #2, 1st Web Audio Conference (http://wac.ircam.fr/), Paris, January 27, 2015 http://wac.ircam.fr/program.html

[10] http://tkita.net/csound/wac2014

[11] WebSocket https://ja.wikipedia.org/wiki/

[12] D3.js Data-Driven Documents http://d3js.org/

[13] http://tkita.net/csound/conf2015