

研究報告

専用アプリ不要の聴衆参加型作品 ”AUDIENCE’S SMARTPHONE JAM SESSION” のシステム構築 SYSTEM CONFIGURATION FOR ”AUDIENCE’S SMARTPHONE JAM SESSION”, AN AUDIENCE-DRIVEN INTERACTIVE PIECE REQUIRING NO DEDICATED APPS

喜多 敏博 KITA Toshihiro
熊本大学 Kumamoto University

概要

iPhone や Android 携帯などのスマートフォンやタブレットを用いて、専用アプリをインストールすることなく即座に 16 名の聴衆がリアルタイム音響生成に参加できる小品のために構築したシステムについて説明する。本システムは、WiFi ルータを用い、それに繋がった Linux ラップトップを中心として構成されている。lighttpd 上で動作する PHP スクリプトに対して、スマートフォン等のブラウザから XMLHttpRequest によるデータ送信を行うことで、OSC メッセージが生成され、Csound での音響生成を制御するものである。

1. はじめに

”Audience’s Smartphone Jam Session” は、バークリー音楽大学で開催された Csound Conference 2013 [1, 2] の 2013 年 10 月 27 日のコンサートで著者が発表した 6 分間程度の作品である。コンサート会場に来た聴衆が、自分の iPhone や Android 携帯などのスマートフォンやタブレットの設定を若干変更するだけで、即座に作品に参加できるようになっており、聴衆がスマートフォンやタブレットをタッチする度に生成される音響がリアルタイムで変化するするように設計した。本稿では、この作品のために構築したシステムについて概説する。

2. 作品の概要

作品 ”Audience’s Smartphone Jam Session” は、聴衆とのライブインタラクションによって完成する作品である。同時に最大 16 人が、iPhone や Android 携帯などのスマートフォンやタブレットを用いてインタラクションに参加することができる。

最初のパート（約 90 秒）は、多数の単純な減衰音から成っていて、聴衆はスマートフォン等をタッチするこ

とで、周波数とリバーブの深さを変化させることができる。

2 番目のパート（約 100 秒）については、聴衆は参加させず、筆者が普段使っている Android 携帯のみで制御した。ゆっくりと波打つ音の塊に対し、Android 携帯を傾けたり、振り回したりすることで、その歪みの度合いや音響定位を変化できるようにしてある。

最後の 3 番目のパート（3 分程度）は、無音状態から始まり、各聴衆が自分のスマートフォン等をタッチして初めて音が発せられ、タッチするタイミングによって、生成される音のパターンが変化する。画面にタッチした聴衆の人数に従って、曲の展開が進められる。

3. システム構成の概要

本作品のために構築したシステムは、図 1 に示すような要素で構成されている。2013 年 8 月 10 日に熊本市現代美術館 [3] で行ったミュージックウェーブ 071 ラップトップミュージック コンサート&ワークショップ [4] でも同一システムを用いた。以下で各要素について概説する。なお、本作品に関連するデータやファイルはすべてウェブ上 [5] で公開している。

3.1. XMLHttpRequest によるデータ送信

参加を希望した聴衆には、

```
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.
```

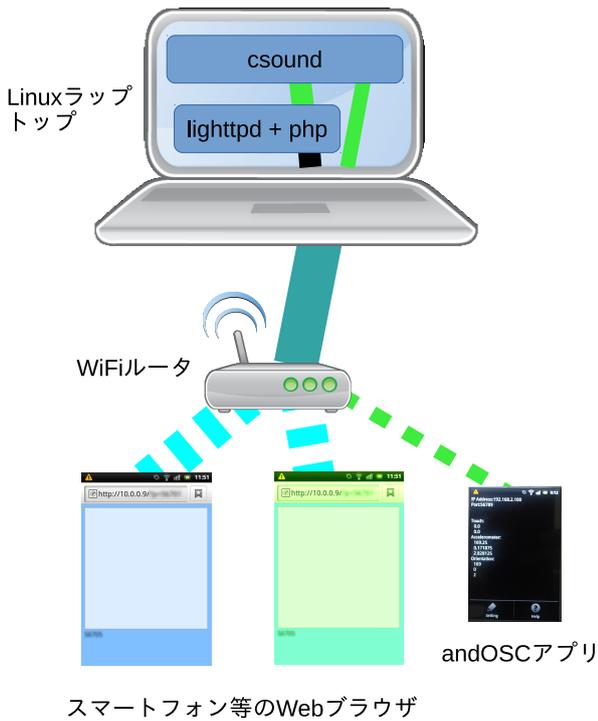


図 1. システム構成

と書かれたカードを配布し、これを見て事前に設定を行ってもらった。それにより、スマートフォン等の Web ブラウザ画面上には、図 1 の下部に示す画面のようなタッチできる領域が表示されるが、これは、図 2 に示すように、HTML5 の canvas 要素を用いて、タッチした点の位置を検出するためのものである。

JavaScript の touchstart というイベントが発生した時に、つまり、聴衆のスマートフォン等の画面がタッチされた時に、touchHandler (図 2 中央付近で定義) という関数が呼び出され、そこで行う処理として、Linux ラップトップへのデータ送信、および、canvas 要素上へ(タッチした点を中心とする)小円の描画が実行される。Linux ラップトップへのデータ送信は、XMLHttpRequest オブジェクトの open メソッドおよび send メソッドを用いて行っている。touchstart イベント検出と XMLHttpRequest による通信を採用したことで、専用アプリのインストール等は不要となり、スマートフォン等のブラウザの画面上での操作ではあるが、今回の用途には十分な低レイテンシーを実現している。

canvas 要素については、その width と height という属性を、図 2 の末尾近くにあるように 300px に固定することで、タッチした点の x 座標と y 座標の値が、デバイスの画面の大きさによらず、最大 300 になるようにした。また、画面が意図しないピンチアウト、ピンチインにより拡大縮小されてしま

```
<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>
```

図 2. 聴衆がタッチする画面の HTML および JavaScript ソース

い、操作が安定的にできなくなることを避けるため、図 2 の冒頭近くにあるように<meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=no"/> と指定した。

3.2. lighttpd と PHP

図 2 の中央近くに、

```
xhr.open("GET", "./oscsend.php?x="+e.touches[0].pageX+
"&y="+e.touches[0].pageY+"&p="+<?php echo $_GET['p'];?>,
true);
```

とあるように、聴衆のスマートフォン等からのデータは、Linux ラップトップ (Ubuntu 13.10) 上に設置された oscsend.php という PHP スクリプトに対して送信される。送信されるデータは、タッチした点の X および Y 座標、各端末に固有の値である。今回のシステムでは、PHP スクリプトは lighttpd [6] という HTTPD 上で動作させた。lighttpd のバージョンは 1.4.31, PHP のバージョンは 5.5.3-1ubuntu2 である。PHP は PHP-FPM を介して lighttpd 上で動作するように/etc/lighttpd/conf-available/15-fastcgi-php.conf の記述を変更するなどして設定した [7]。lighttpd は、汎用的に広く用いられている HTTPD である Apache と比較して、メモリ消費量が少なく、CPU 負荷の少ない高速動

作をすることを意図して開発されたものである [8]。

データを受け取った `oscsend.php` は、`OSC.php` [9] で定義されているクラスを用いて、受け取ったデータを OSC メッセージに変換して、音響を生成するプロセス (今回の場合 Csound) に送信する。

3.3. OSC メッセージの形式

`andOSC`[10] という無料の Android アプリが生成する OSC メッセージは、スマートフォン内蔵の各種センサーによって得られる情報に基づいており、

- 1) Touch(x,y)
- msg : /touch 0 0
- 2) Accelerometer
- msg : /acc 0 0 0
- 3) Orientation
- msg : /ori 0 0 0

のような形式である。1), 2), 3) はそれぞれ、画面上のタッチした場所、加速度の値、向いている方向によって決定される値である。今回、これを参考として OSC メッセージの形式を決めた。そのため、今回の作品には、スマートフォン等のブラウザ経由だけではなく、`andOSC` アプリを用いても参加することができる。

なお、2 番目のパートについては、`andOSC` がインストールされた Android デバイスからの OSC メッセージのみを受信することが前提となっており、1) Touch のデータと 3) Orientation のデータを用いている。

音響生成には、Csound [11, 12, 13] を用いており、`lighttpd` が動作しているのと同じ Linux ラップトップ上で動作させた。`OSClisten` [14] などのオブコードを用いて受信した OSC メッセージに基づき、2 節で述べたように、生成される音響がリアルタイムで変化するように設計した。詳しくは Web 上に公開している CSD ファイル [5] を参照されたい。

4. おわりに

タッチパネルを採用したデバイス (タッチイベントがブラウザで検出されるデバイス) を持っていれば、特別なアプリ等をインストールする手間が不要で、WiFi 接続さえできれば、聴衆が OSC メッセージ生成してそれを介して作品に参加することのできる汎用性のあるシステムが構築できた。また、本システムは、OSC 準拠のツールやソフトであれば、Csound に限らず適用可能である。

現時点では聴衆とのインタラクションの結果が反映されるのは、音響のみであるが、ビジュアル要素にも反映させることや、個々の聴衆へのフィードバックがよりわ

かりやすく認識できるように改善することを考えたい。

5. 参考文献

- [1] <http://csound.org/>
- [2] Csound Conference 2013 Schedule
<http://www.csounds.com/wp-content/uploads/2013/10/CsoundConferenceSchedule10-23-13.pdf>
- [3] 熊本市現代美術館 | CAMK
<http://www.camk.or.jp/>
- [4] 熊本市現代美術館 CAMK Blog
<http://camk.glide.co.jp/blog/index.php?eid=1716>
- [5] <http://tkita.net/csound/conf2013/>
- [6] Home - Lighttpd <http://www.lighttpd.net/>
- [7] 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>
- [8] lighttpd - Wikipedia
<http://ja.wikipedia.org/wiki/Lighttpd>
- [9] Open Sound Control for PHP
<http://opensoundcontrol.org/implementation/open-sound-control-php>
- [10] `andOSC`
<https://play.google.com/store/apps/details?id=cc.primevision.andosc>
- [11] Csound on Sourceforge
<http://csound.sourceforge.net/>
- [12] Csounds.com <http://www.csounds.com/>
- [13] Csound - FLOSS Manuals,
<http://en.flossmanuals.net/csound/>
- [14] `OSClisten`
<http://www.csounds.com/manual/html/OSClisten.html>

6. 著者プロフィール

喜多 敏博 (KITA Toshihiro)

1967 年に奈良に生まれる。京都大学大学院工学研究科博士後期課程研究指導認定退学、熊本大学 工学部助手、総合情報基盤センター准教授、e ラーニング推進機構教授、現在に至る。工学博士 (名古屋大学, 2005 年)。e ラーニングシステム、非線形システム、電子音楽に興味を持つ。<http://tkita.net>