

visual performance workshop

青木 聖也

多摩美術大学情報デザイン研究室

August 18, 2017

Contents

まずはじめに

インターフェースを作る

OSC

信号のやりとり

touchOSC

touchOSC から openFrameworks への OSC 送信

オーディオリアクティブ

FFT

音を描画描画パラメータに割り当てる

しきい値を超えたら展開が進む

資料について

今回の資料を以下に公開しています

- ▶ 全体説明: <http://scottallen.ws/tamabi/vjworkshop>
- ▶ プログラム: <https://github.com/5c0tt411en/iddvjworkshop2017>
- ▶ スライド:
<https://github.com/5c0tt411en/iddvjworkshop2017/blob/master/Slide/>

OSC とは

Open Sound Control の略アプリ間の通信に使用する形式

- ▶ URL 形式の名前付け
- ▶ 数値やシンボルなど様々な信号を伝達可能
- ▶ 遅延少ない

OSC の使用事例 1 マシン間での映像同期

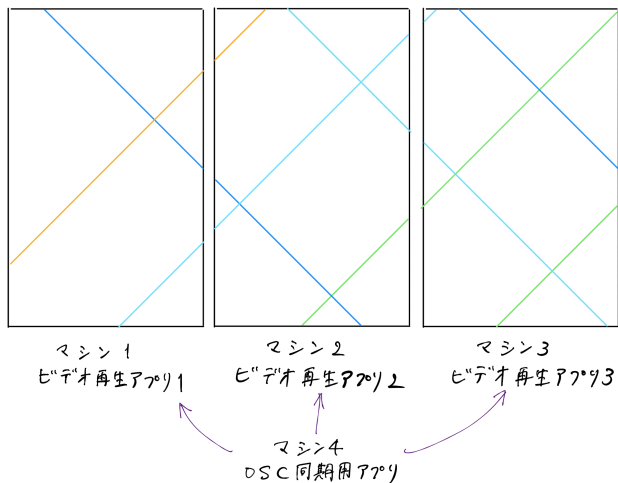


Figure 1: マシン間での映像同期参考図

OSC の使用事例 2 アプリ間で座標を渡す

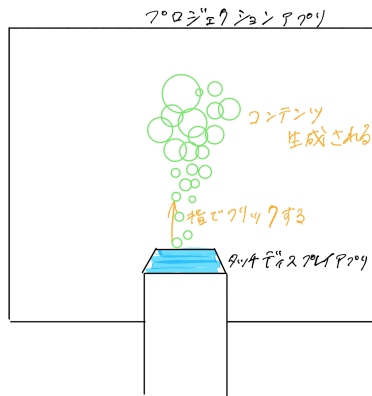


Figure 2: アプリ間で座標を渡す参考図

oF - oF の OSC 通信について

oF - oF の OSC 通信に関しては以下にサンプルを公開しています

- ▶ 全体説明: <http://scottallen.ws/tamabi/summerworkshop2017>
- ▶ プログラム: <https://github.com/5c0tt411en/iddsummerworkshop2017>
- ▶ スライド:
<https://github.com/5c0tt411en/iddsummerworkshop2017/blob/master/Slide/day2>

OSC を使った通信方法

送信側

- ▶ 相手側マシンの ip アドレスを指定: 169.254.11.14
- ▶ 通信用ポートを指定: 8888
- ▶ アドレスを指定: /stat
- ▶ 型を明示して値を送信

受信側

- ▶ 通信用ポートを指定: 8888
- ▶ 型を合わせて値を受信

touchOSC



Figure 3: touchOSC 画面
(<https://hexler.net/software/touchosc>)

touchOSC の利点

- ▶ 手持ちの iOS, Android がインターフェースになる
- ▶ PC で自由にインターフェースを作成できる
- ▶ iOS アプリを開発する必要がない
- ▶ シンプルなので短時間でインターフェースが作成できる
- ▶ コスト 600 円

touchOSC を使ったコントロール

WS01/bin/touchOSC01.touchosc

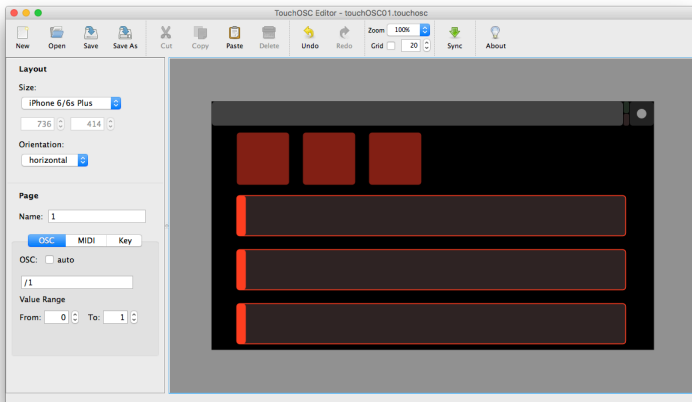


Figure 4: touchOSC01.touchosc 画面

touchOSC を使ったコントロール

WS01/

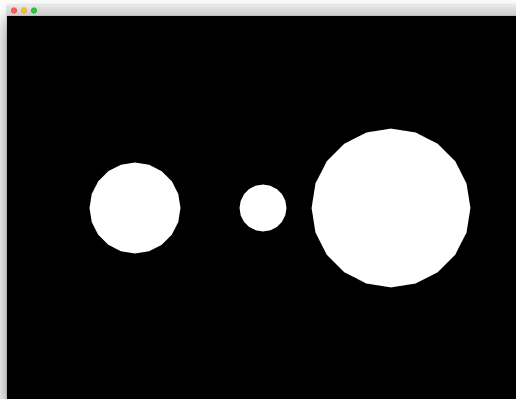


Figure 5: oF scene 1

touchOSC を使ったコントロール

WS01/

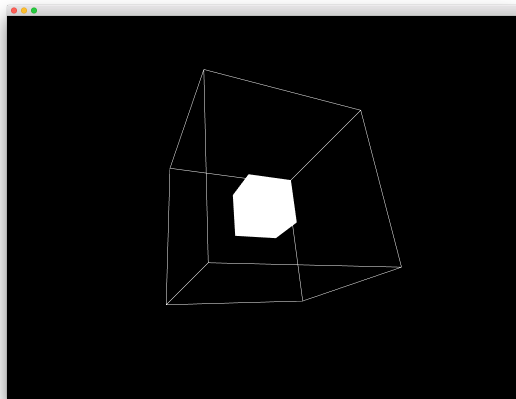


Figure 6: oF scene 2

touchosc を使ったコントロール

WS01/

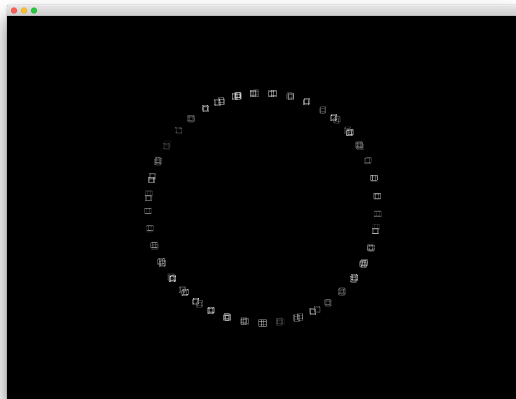


Figure 7: oF scene 3

FFT とは

高速フーリエ変換（Fast Fourier Transform）の略 音の解析
に使用される

- ▶ 音の高さ（周波数）ごとのレベルがわかる
- ▶ サウンドビジュアライズが容易にできる
- ▶ アルゴリズム自体は理解せずとも使用できる
- ▶ Generative な VJ は全員使っている

FFT 高速フーリエ変換 ofxEasyFft

WS02/

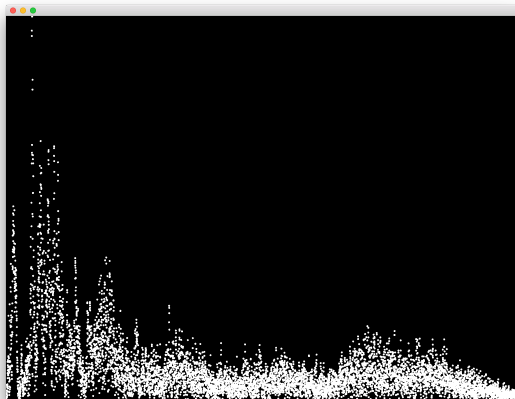


Figure 8: oF easyfft

FFT 高速フーリエ変換 ofxProcessFft

WS03/



Figure 9: of processfft

FFT 高速フーリエ変換 ofxProcessFft

WS04/

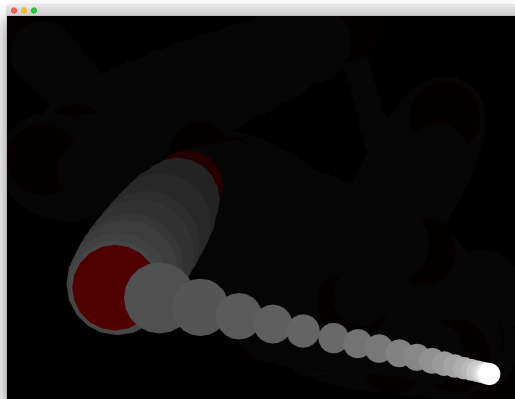


Figure 10: oF processfft

FFT 高速フーリエ変換 ofxProcessFft

WS04/

$$\text{各経過時間} = \text{アプリを立ち上げてからの経過時間} - \text{タイムスタンプ}$$



Figure 11: time の考え方

FFT 高速フーリエ変換 ofxProcessFft

WS04/

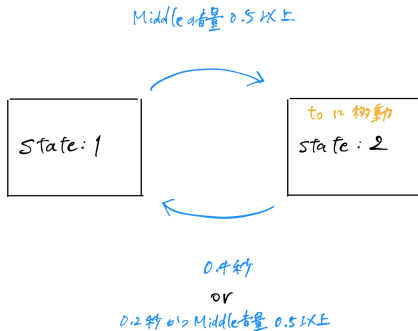


Figure 12: state の移行図

FFT 高速フーリエ変換 ofxProcessFft

WS04/

$now = easeOutCubic(\text{float } t, \text{float } b, \text{float } c, \text{float } d)$

0から
経過時間

初動前の
端角

初動の
差/分

かける
時間

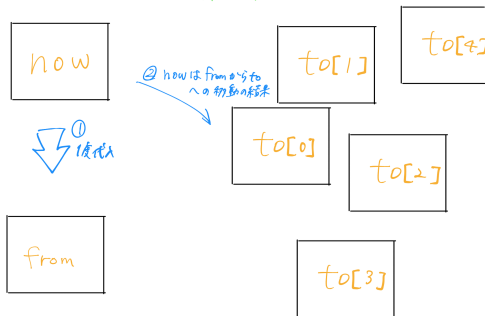


Figure 13: easing 関数の考え方