



Atsushi Toba, Dec 2017

かんたん  
15分!

# JavaScriptで扱う モーションデータ

# INDEX

## 今日の内容

①

Leap Motion概要

②

使い方

③

所感

④

付け足し



# introduction

鳥羽 敦

2016年1月入社

DTP、デザインもやってました

今はFEやってますjs書いてます

:D





# 1. Overview

Leap Motion概要



## Features

## Leap Motionとは？

2012年Leap Motion社から発売、  
3Dモーションキャプチャデバイスの1つ。

MSのKinectが体全体のジェスチャーの認識する  
のに対して、  
Leap Motionの特徴は…

- 手のひらサイズでコンパクトなデバイス
- キャプチャ範囲が60cmぐらい、指先から肘ぐらいまでのジェスチャーを認識
- VR向けの開発もできる (ORION BETA)







# Development

開発環境

## 1. Overview

SDKもいろいろある。

V2



C++



C#



Unity



Objective-C



Java



Python



JavaScript



Unreal Engine [↗](#)

ハードはそのまま、2014年にv2がリリースされより精度がよくなったよ！

## Receivable

どんな情報が使える？

## ● 手

- ・ 位置 (座標x/y/z)
- ・ 右・左
- ・ 向き
- ・ グラブ (握る)

## ● 指

- ・ 位置 (関節 x/y/z)
- ・ 親指～小指
- ・ 曲げ伸ばし
- ・ ピンチ (摘む)

## ● ジェスチャー

- ・ 回転
- ・ スワイフ
- ・ キータップ
- ・ スクリーンタップ

## ● ツール

- ・ 白い棒状の何かを認識



## 2. Usage

使い方





# How to Use

# 使い方

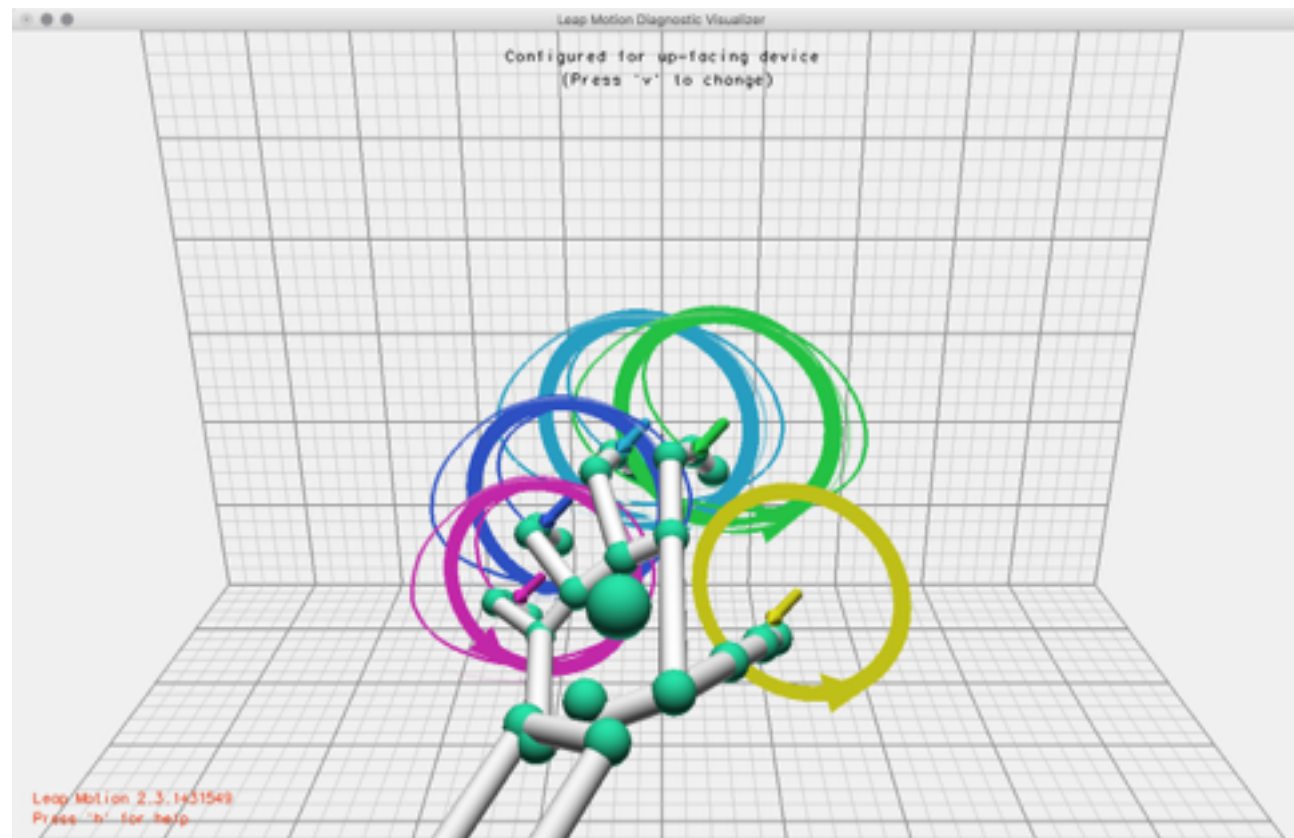
## 使い方

1. ソフトをダウンロードしインストール

(※デベロッパーインストーラはアカウントが必須)

2. PCに繋げる

以上





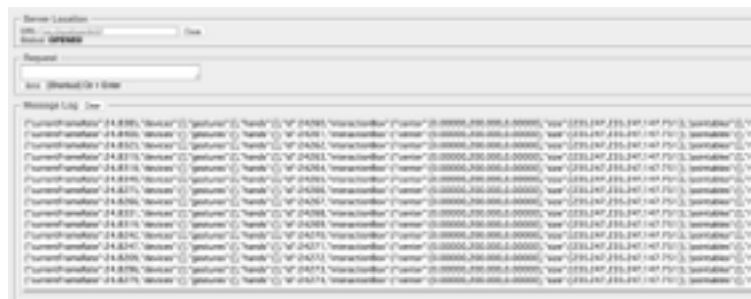
# LEAP meets JS

JavaScriptで使う

Leap Motionを繋げるとサーバが立ち上がり、何をしなくても WebSocket通信 (ws://localhost:6437) が行われている。

モーションの取得情報は25~200fps（毎秒フレーム）でレスポンスが JSON形式のデータで返ってくる（USB3.0接続時）

こんな感じ…。



このレスポンスで得られるデータを元に開発ができる。



# LEAP meets JS

JavaScriptで使う

もう1つの使い方。

もっと手っ取り早い使い方として、WEB向けにデータを解析してくれるオープンソースのAPIが公開されている。

その名も、

# LEAPJS



# leap.js

# JavaScriptで使う

ブラウザで使う場合はDOMに読み込ませるだけで、トラッキングデータをLeapMotionのframeオブジェクトを使って、イベントループ処理ができるようになる。

```
<script src="//js.leapmotion.com/leap-0.6.4.min.js"></script>
```

ライブラリを読み込む使い方と一緒にかんたん！





# leap.js

# JavaScriptで使う

## 処理の書き方

```
Leap.loop({enableGestures: true}, function(frame){  
  // フレームごとの処理を記述  
});
```

enableGestures: true でジェスチャー操作を有効にする。



## 2. Usage

leap.js | JavaScriptで使う

実際ブラウザで情報が取得できているかどうか見てみる。

<http://localhost:3000/first.html>



# leap.js | JavaScriptで使う

インスタンス、イベントバインド形式で書くことも可能

```
// LeapMotion Class
var controller = new Leap.Controller({enableGestures: true});

controller.on('frame', function(frame) {
  // フレームごとの処理を記述
});
controller.connect(); // 接続
```

トラッキングデータは、コールバック関数に渡される引数(frame)に格納される。



# leap.js | JavaScriptで使う

frame中に手、指の座標、動き、角度など全ての情報が入ってくる。  
ここから欲しい情報を引き出して使う。(Frameオブジェクトを使う)

```
frame.hands // 手のデータ (配列)  
frame.fingers // 指のデータ (配列)  
frame.gestures // ジェスチャーのデータ (配列)
```

例えば、指の本数を取得するとき。

```
Leap.loop({enableGestures: true}, function(frame){  
  console.log(frame.fingers.length);  
});
```



# leap.js | JavaScriptで使う

ジェスチャー(frame.gestures)情報の取得はtypeプロパティを参照。  
用意されているtypeは以下の4つ。

● circle      ● swipe      ● keyTap      ● screenTap

stateプロパティというものもあり、これはジェスチャーの開始、途中、終了に対応した文字列が入っている。

● start      ● update      ● stop





## 2. Usage

leap.js | JavaScriptで使う

こんなの作ってみた。

<http://localhost:3000/second.html>

<http://localhost:3000/fever.html>



## 3. Impression

所感



Impression

使ってみた感想

## Good

- VRの開発に少し便乗した感
- 映画のようなUI設計を夢みて少しモチベーション上がる
- 元Flasherの方、three.jsやCreateJSを使ってガシガシ作っていそう
- 安価で手軽に始められる

## Bad

- 処理の重さ、誤認識を克服するにはそれなりにノウハウが必要な感じ
- ゴリラアーム問題を少し実感

# Impression

使ってみた感想



## 4. An Addition

ちょっと付け足し





# 3DCG

# three.jsとCannon.js

さらにLeap Motionに3D描画を加えてみる。

- three.js

WebGLを使った3D表現がJavaScriptでかんたんに扱えるライブラリ

- Cannon.js

物理演算を提供してくれるいわゆる「物理エンジン」ライブラリ



3DCG

three.jsとCannon.js

Leap Motionのジェスチャーで剛体の衝突を制御してみる

[https://tobaatsushi.github.io/three\\_cannon/](https://tobaatsushi.github.io/three_cannon/)

# Reference Materials

## 参考リンク

- Leap Motion

<https://www.leapmotion.com/>

- LeapJS

<https://developer-archive.leapmotion.com/javascript>

- Gallery Example

[developer-archive.leapmotion.com/gallery/tags/javascript](https://developer-archive.leapmotion.com/gallery/tags/javascript)


- [新MacBook Airでタッチ画面が断念された理由 \[WIRED\]](#)

- three.js

<https://threejs.org/>

- Cannon.js

<http://www.cannonjs.org/>

A black and white photograph of a person from behind, wearing a quilted jacket and a hood, looking out at a sunset over a forest. The sun is low on the horizon, creating a bright glow. The trees are silhouetted against the light sky.

Enjoy your  
JS LIFE!

ご清聴ありがとうございました