# Leap Motion

# 2008
# Michael Buckwald et David Holz

Dispositif de reconnaissance de mouvement des mains, créé par Leap Motion, Inc.

# 2012 - v1 (bêta)

Le produit est majoritairement destiné aux développeurs, dans le but d’être testé et amélioré constamment.

# 2013 - version commercialisée

Le contrôleur devient accessible au grand public, et offre déjà une large galerie de démos.

# 2013

- HP propose un laptop intégrant le contrôleur Leap Motion : le **HP Envy Leap Motion SE**.
- Asus sort également un modèle livré avec le Leap : le **Asus N750JV-T4169H**

# 2014 - v2

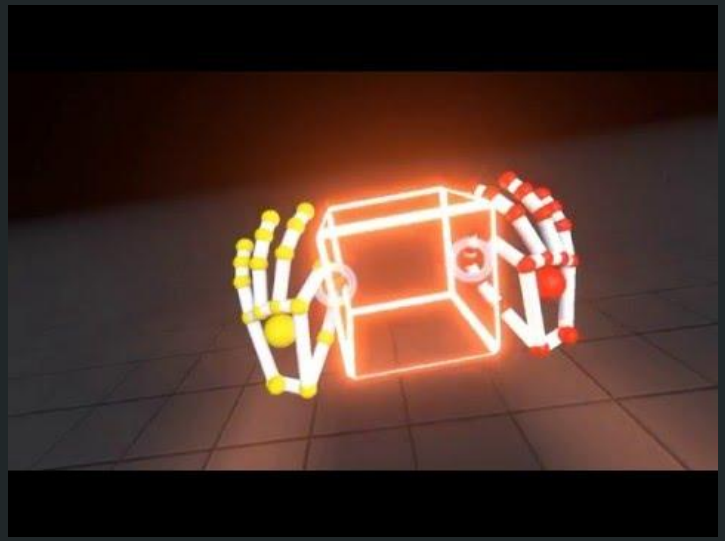
Amélioration de la reconnaissance des paumes/doigts/phalanges, meilleure gestion de la lumière, meilleure précision de détection.

OSVR


# 2015

Leap Motion s’associe avec le projet OSVR (Open Source Virtual Reality) et permet l’intégration du boîtier sur le Razer OSVR.

# Et maintenant ... ?



Leap Motion se focalise beaucoup sur la VR, avec récemment la sortie de la plate-forme VR : **Orion** (bêta)

Unity3d & Unreal Engine

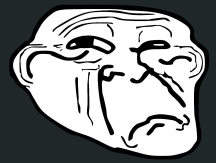
# Leap Motion Visualizer

# JavaScript

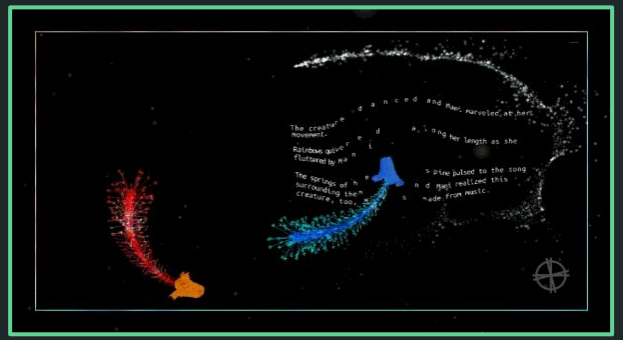
Leap Motion dans le navigateur ?

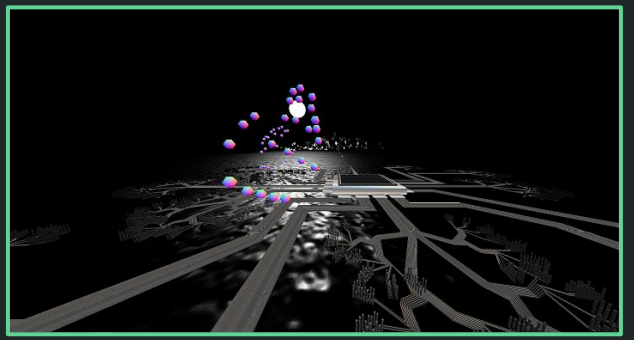
https://developer.leapmotion.com/javascript

v0.6.4 depuis 2015 !

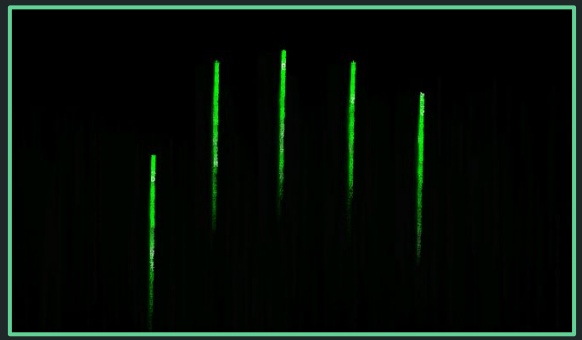


# Exemples / Démos full-web



# Exemples / Démos full-web




Et d’autres par ici : https://developer-archive.leapmotion.com/gallery/tags/javascript

# Getting started

https://developer.leapmotion.com/getting-started/javascript

# Installation

```
$ npm install leapjs -v0.6.4
```

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

# Initialisation

```
const controller = new Leap.Controller();
controller.connect(); // Ouvre la connexion WebSocket

controller.on('frame', (frame) => {
    console.log(`Frame event for frame ${frame.id}`);
});
```

Codepen

# Interagir

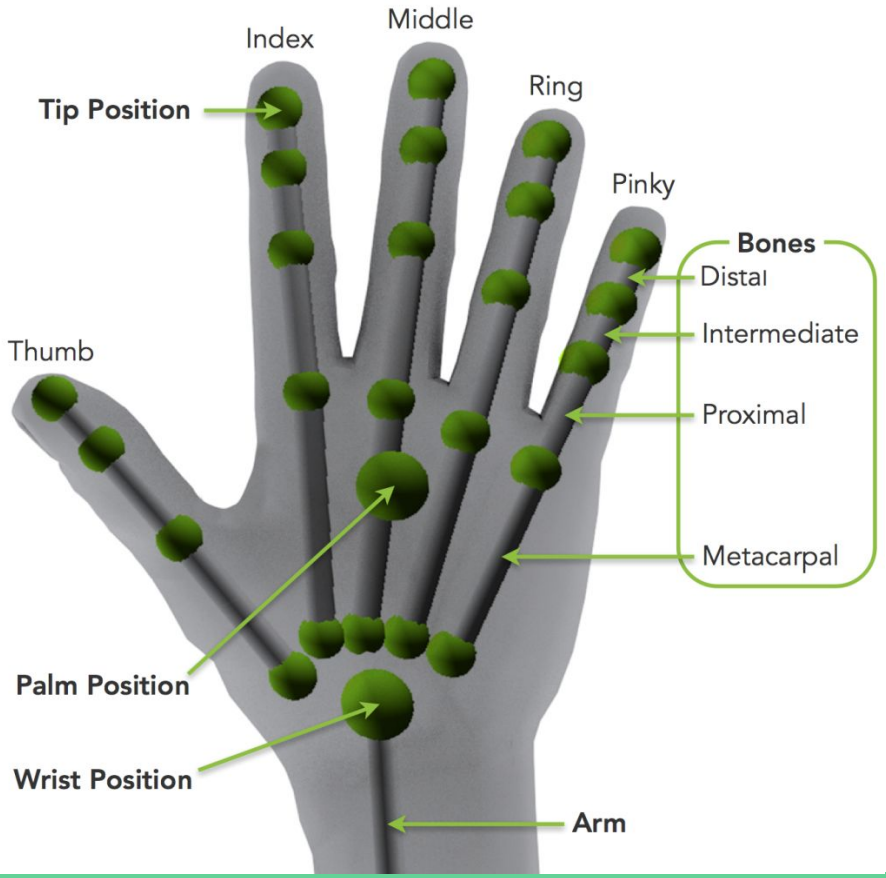
# Hands / Fingers

```
controller.on('frame', (frame) => {

    frame.hands // Tableau contenant des données sur les mains

    frame.hands[].fingers[] // Tableau de doigts pour chaque main

});
```

Codepen

# Dessiner

# Anatomie d’une main



# Récupérer les coordonnées de la paume de la main

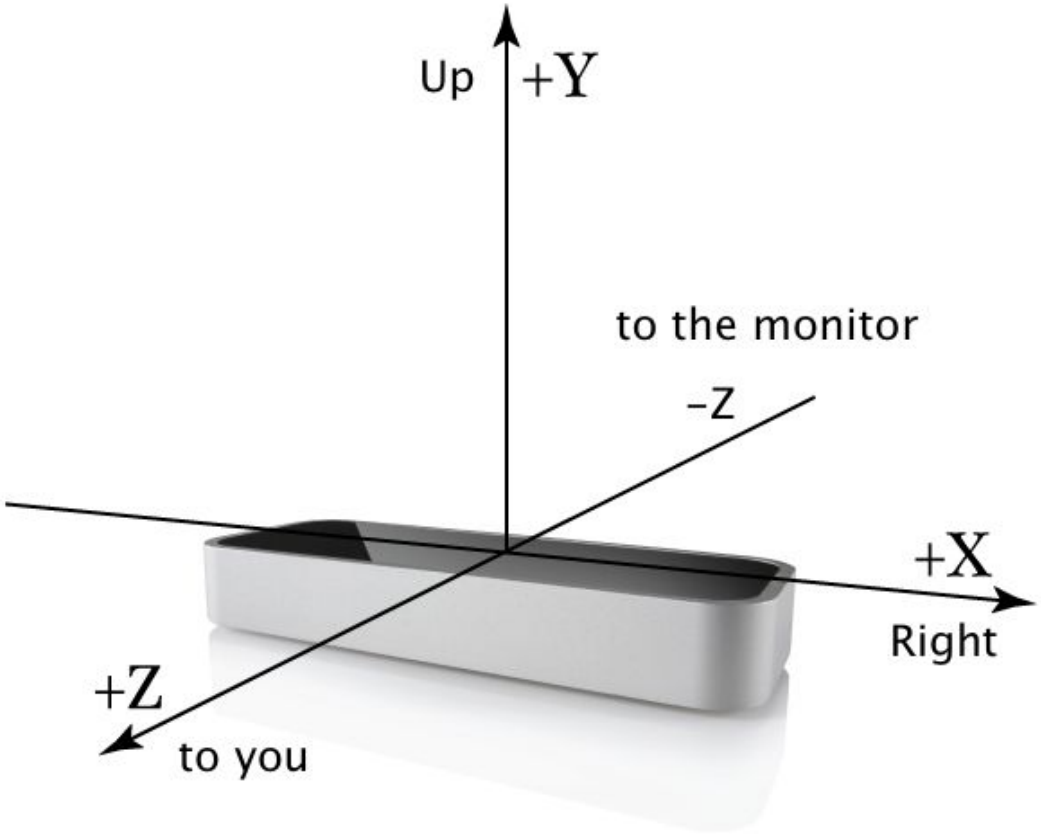
```
controller.on('frame', frame => {

  frame.hands.forEach( hand => {
    hand.palmPosition[0] // x
    hand.palmPosition[1] // y
    hand.palmPosition[2] // z
  });

});
```

# Récupérer les coordonnées des doigts

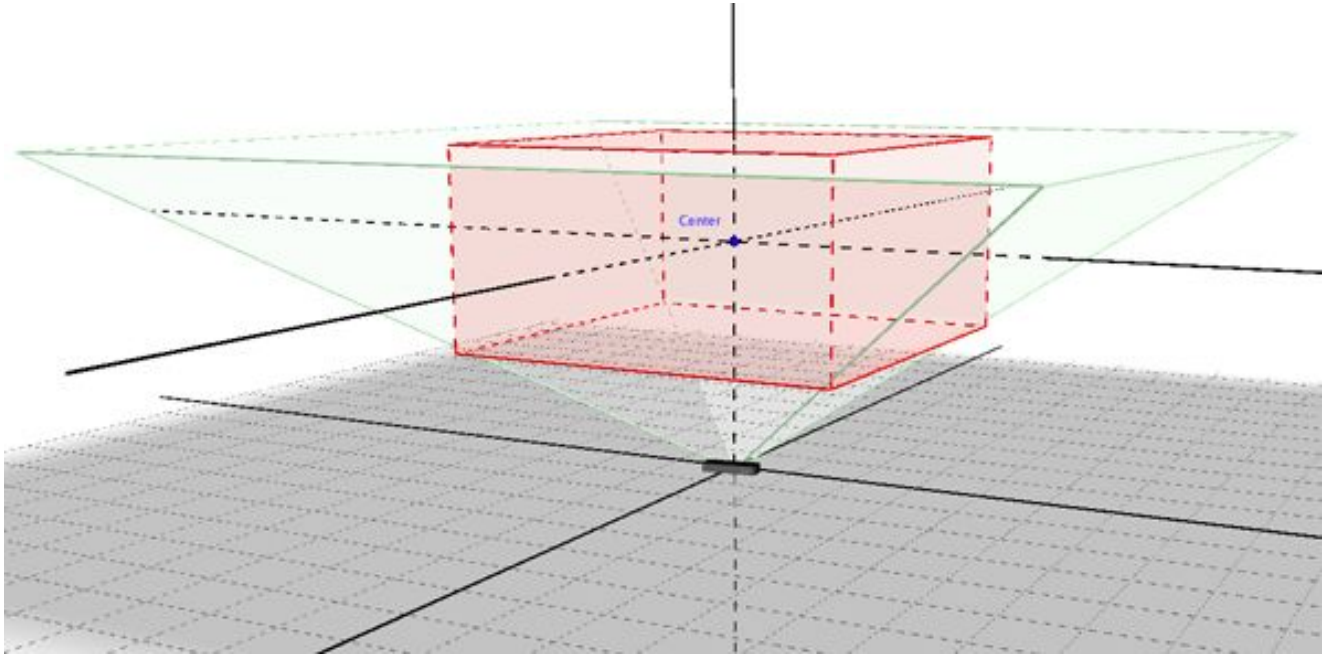
```
frame.hands.forEach( hand => {
 hand.fingers.forEach( finger => {
   finger.carpPosition // carpal
   finger.mcpPosition // metacarpal
   finger.pipPosition // proximal
   finger.dipPosition // intermediate phalange
   finger.tipPosition // distal phalange
 });
});
```

Codepen

# Système de coordonnées Leap



# Interaction box



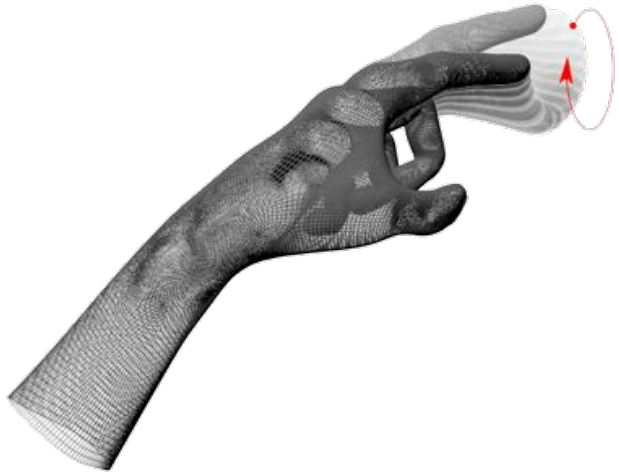

Disponible dans l’objet : `frame.interactionBox`

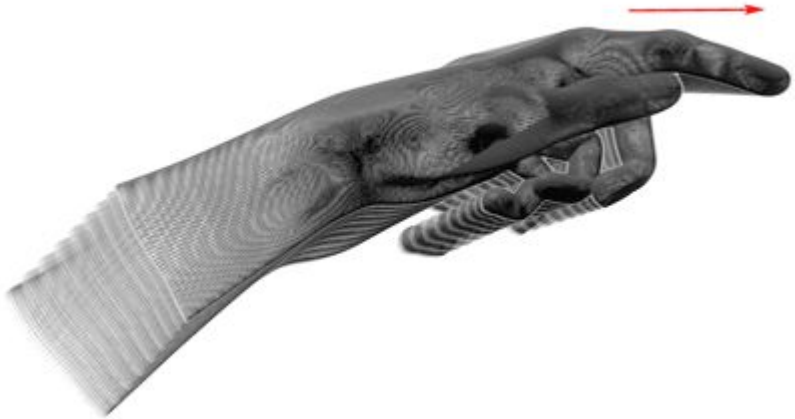
# Gérer des coordonnées 3D dans un système 2D

```
function getCoords(leapPoint, frame, canvas) {
    const iBox = frame.interactionBox;
    const normalizedPoint = iBox.normalizePoint(leapPoint, true);

    return {
        x : normalizedPoint[0] * canvas.width,
        y : (1 - normalizedPoint[1]) * canvas.height
    };
}
```

# Gestures

Reconnaître un mouvement de main de l'utilisateur





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

controller.on('frame', frame => {
  frame.gestures.forEach(gesture => {
    switch (gesture.type) {
      case 'circle'    : console.info('Circle detected'); break;
      case 'keyTap'    : console.info('KeyTap detected'); break;
      case 'screenTap' : console.info('ScreenTap detected'); break;
      case 'swipe'     : console.info('Swipe detected'); break;
    }
   });
});
```

Codepen

# LeapJS plugins

Collection de plugins utilisables pour le Leap Motion Controller

```
<script src="https://js.leapmotion.com/leap-plugins-0.1.12.js"></script>
```

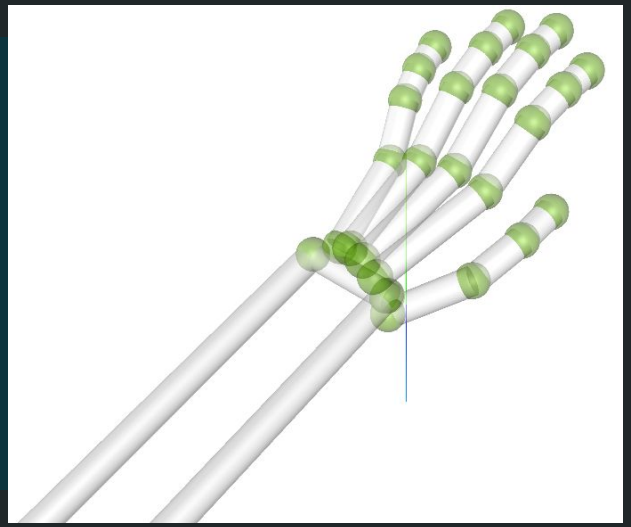
# *screenPosition*

```
Leap.loop({

  hand: function(hand){
    console.log( hand.screenPosition() ); // returns [156,204,0]
  }

}).use('screenPosition');
```

http://leapmotion.github.io/leapjs-plugins/docs/#screen-position

# *boneHand*

```
Leap.loop()
  .use('boneHand', {
    targetEl: document.body,
    arm: true
  });
```



http://leapmotion.github.io/leapjs-plugins/main/bone-hand/

# … et d’autres

- Hand Entry Emit events when a hand enters of leaves the field of view.
- Hand Hold Save data on to hands or fingers which will be persisted between frames.
- Version Check Ensure a minimum protocol version when running your app.
- Playback Record hand-data from the Leap, compress it, and use it to animate your app.
- Transform Translate, rotate, and scale Leap Motion data. Easily.

https://github.com/leapmotion/leapjs-plugins

# LeapJS widgets

Aide à la manipulation de surfaces 2D dans un espace 3D