

# **Dispositif technique**

**Annexe**

## Sommaire

0 4

### Introduction

0 5

### 1 / Hacking Of

1 2

### 2 / Constat

1 8

### 3 / Poursuite

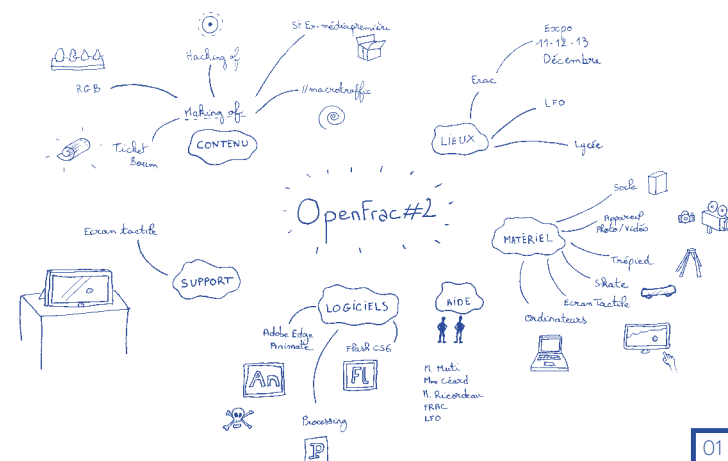
# 1 /

## Hacking Of

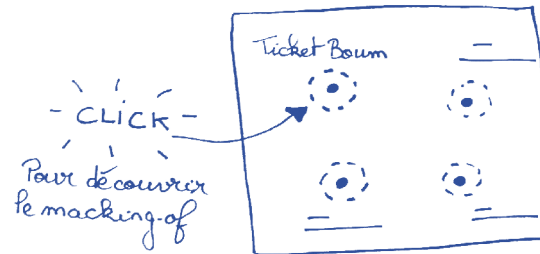
La vidéo interactive réalisée sur Flash CS6 s'intitule Hacking Of. Le but est de présenter au public un making of de l'exposition OpenFrac#2<sup>1</sup>. Ce making of étant interactif, l'utilisateur aura le choix d'approfondir certaines parties de la vidéo<sup>2</sup>, pour avoir des informations supplémentaires, en cliquant sur des boutons interactifs.

### Introduction

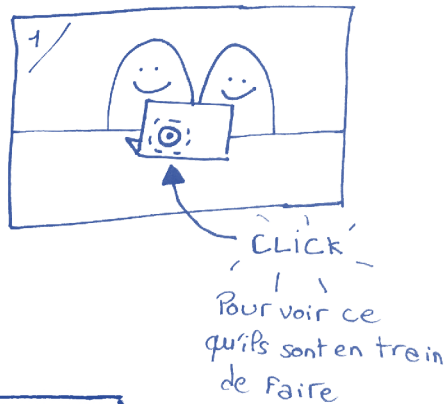
Nous avons réalisé un projet avec le FRAC PACA qui s'intitule OpenFrac#2. Ce projet a pour but de faire participer le public du FRAC à la médiation de l'exposition « Histoires Parallèles ». Nous étions en collaboration avec l'équipe du FRAC, nos enseignants et avec l'association marseillaise Réso-nance Numérique. C'est ainsi que j'ai été en charge de la réalisation d'une vidéo interactive qui permettra de garder une trace de l'événement et de sa préparation, d'en faire la promotion et d'aider à la compréhension des projets d'augmentation proposés. À l'aide de mon équipe, cela m'a permis de prototyper, pour la toute première fois, une vidéo interactive. Par avant, j'ai réalisé plusieurs stages pour me former sur les techniques de vidéo. Je vous renvoie donc vers mon rapport de stage pour découvrir ce que j'ai pu y apprendre.



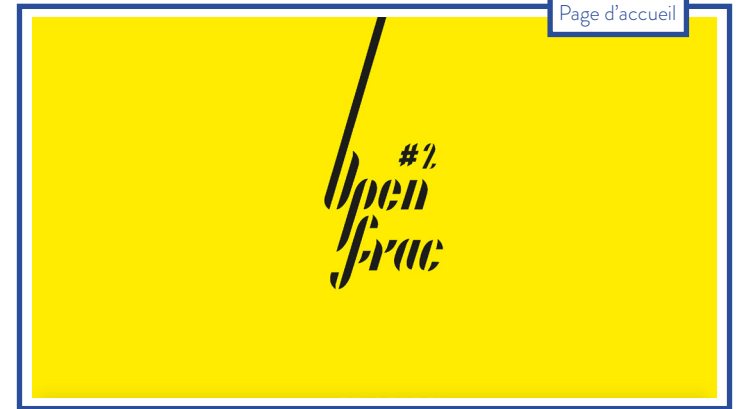
01  
Carte du projet  
OpenFrac#2



Bouton  
Interaction



↳ plan supplémentaire  
suite au click



Clique sur le logo pour lancer la  
vidéo interactive.

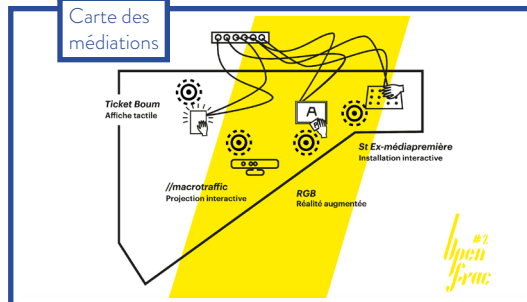
#### Code

```
// stop l'animation générale
stop();
```

```
//action sur bouton openfrac pour aller au menu
principal
openfrac.onRelease = function() {
    gotoAndPlay(«menu»,1);
}
```

```
// cacher curseur ou pointeur
//Mouse.hide();
```

02

Carte des  
médiations

Clique sur une des œuvres pour découvrir le déroulement du travail réalisé.

### Code

```
// stop l'animation générale
stop();

// musiques
music1 = new Sound();
music1.attachSound(«fantome_CMJN.mp3»);
music2 = new Sound();
music2.attachSound(«podington_bear_black_eyed_susan_no_future.mp3»);
music3 = new Sound();
music3.attachSound(«new_life_at.mp3»);
music4 = new Sound();
music4.attachSound(«pantystep_kinect.mp3»);

//action sur bouton AT pour lire le making of affiche tactile
AT.onRelease = function() {
  gotoAndPlay(«affichetactile1»,1);
}

music3.start();

//action sur bouton kinect pour lire le making of kinect
kinect.onRelease = function() {
  gotoAndPlay(3);
  music4.start();
}

//action sur bouton cmjn pour lire le making of réalité augmentée
cmjn.onRelease = function() {
  gotoAndPlay(25);
  music1.start();
}

//action sur bouton No Future pour lire le making of installation interactive
nofuture.onRelease = function() {
  gotoAndPlay(17);
  music2.start();
}
```

03

Narration  
de la vidéo

Clique sur le rond pour obtenir une vue rapprochée de l'objet manipulé par la jeune femme.

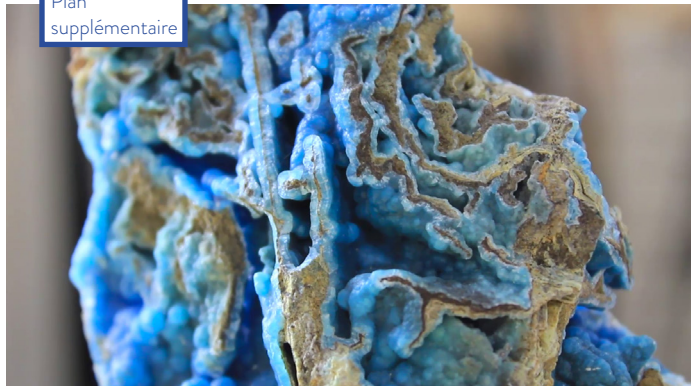
### Code

```
// stop l'animation générale
stop();

//action sur bouton_interaction_kinect_1_1
bouton_interaction_kinect_1_1.onRelease = function() { //ajout d'une fonction sur le bouton interaction nommé bouton_interaction_kinect_1_1 placé dans le montage flash
  gotoAndPlay(4); //bouton interaction appuyé, on se déplace alors sur la frame 4 du montage flash. Sur la frame 4 se situera le plan supplémentaire}
  // bouton_interaction_kinect_1_1 // bouton_interaction_kinect_1_1.
  visible = false; //permet de nommer l'état d'un bouton interaction, true ou false, correspondant ici à visible ou invisible
  setTimeout(myFunction_kinect_A, 7000); //initier un timer, qui permet ici de faire apparaître des éléments à un moment précis. Ici création d'un
```

```
timer A, qui permettra de faire apparaître un élément au bout de 7000ms
function myFunction_kinect_AO { //attribuer la fonction du timer A à un élément bouton_interaction_kinect_1_1.
  visible = true; //permet de rendre visible le bouton interaction en suivant la fonction du timer A, c'est à dire le bouton sera visible après 7000ms}
  setTimeout(myFunction_kinect_B, 11000); // initier un timer, qui permet ici de faire disparaître des éléments à un moment précis. Ici création d'un timer B, qui permettra de faire disparaître un élément au bout de 11000ms
function myFunction_kinect_BO { //attribuer la fonction du timer A à un élément bouton_interaction_kinect_1_1.
  visible = false; //permet de rendre invisible le bouton interaction en suivant la fonction du timer B, c'est à dire le bouton sera invisible après 11000ms}
```

04-1

Plan  
supplémentaire

Plan supplémentaire dans la vidéo si bouton interaction appuyé.

04-2

Suite  
de la vidéo

Suite de la vidéo si bouton interaction non appuyé.

05

Crédits



Clique sur menu pour retourner au menu.

### Code

```
// stop l'animation générale
stop();
music4.stop(); //la musique de la vidéo s'arrête
//action sur bouton retour menu
retour_menu_kinect.onRelease = function() { //
ajout d'une fonction lorsqu'on appuie sur le bouton
retour_menu
    gotoAndPlay(2); // on revient en frame 2 quand
le bouton retour_menu est appuyé, qui correspond
à la frame où se situe le menu
}
```



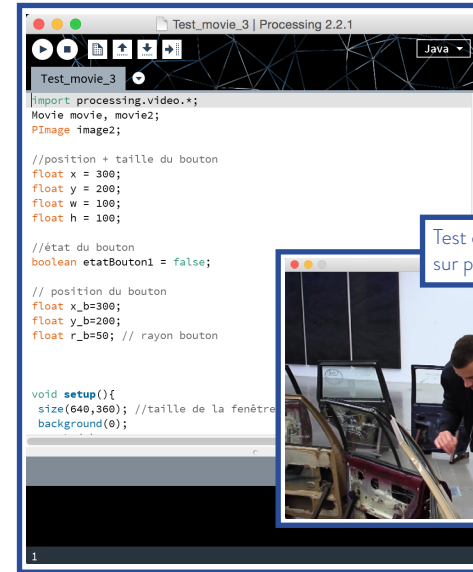
OpenFrac#2,  
nocturne



2 /

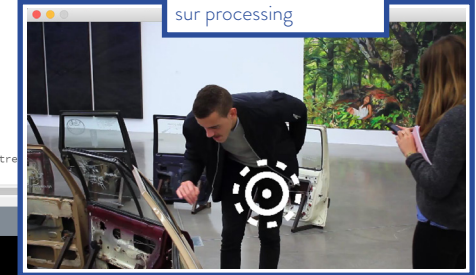
## Constat

Le constat que je peux tirer de cette expérience est que le logiciel Adobe Edge Animate n'est pas adapté pour ce type de vidéo interactive. En effet, en ayant fait des recherches auparavant, je pensais que ce logiciel nous aurait aidé afin de réaliser l'interaction. Or, il s'avère qu'il ne correspond pas à ce trop gros format de vidéo interactive. De ce fait, nous avons utilisé le logiciel Adobe Flash CS6, qui nous a permis de rendre la vidéo interactive stable. Cependant, certains bug liés aux « timer » persistent. Le but serait donc de réaliser la vidéo interactive sur Processing<sup>3</sup>, afin de stabiliser complètement la vidéo et de la rendre open source.



03

Test de vidéo interactive  
sur processing



## Code

```
//importer librairie
import processing.video.*;
//importer une la librairie vidéo import ddf.minim.*;
//importer une librairie audio //déclaration des variables
Movie movie1, movie1_1, movie1_2, movie1_3, movie1_4;
//déclarer les différentes vidéos PImage image1, image2, image3,
image4;
//déclarer les différentes images Minim minim;
AudioPlayer song;
//position + taille du bouton
float x = 300;
// espace à gauche du bouton float y = 200;
//espace en dessus du bouton float w = 100;
//largeur du bouton float h = 100;
//hauteur du bouton //état initial du bouton
boolean etatBouton1 = false;
//attribution de l'état initial du bouton 1 false boolean etatBouton2 =
false;
```

```
//attribution de l'état initial du bouton 2 false boolean etatBouton3 =
false;
//attribution de l'état initial du bouton 3 false boolean etatBouton4
= false;
//attribution de l'état initial du bouton 4 false // position du bouton 1
float x_b1=300;
float y_b1=450;
float r_b1=50; // rayon bouton
// position du bouton 2
float x_b2=600;
float y_b2=450;
float r_b2=50; // rayon bouton
// position du bouton 3
float x_b3=600;
float y_b3=550;
float r_b3=50; // rayon bouton
// position du bouton 4
float x_b4=800;
float y_b4=550;
float r_b4=50; // rayon bouton
void setup(){
// son
minim = new Minim(this);
song = minim.loadFile(«fantome_cmjn.mp3»);
//charger un son size(1440,816); //taille de la fenêtre
background(0);
//couleur de l'arrière plan stroke(0);
noFill();
// Load and play the video in a loop
//Vidéo 1
movie1 = new Movie(this, «1.mp4»);
//charger une vidéo nommée «1.mp4» movie1.loop();
//lire en boucle //Vidéo 1_1
movie1_1 = new Movie(this, «1_1.mp4»);
movie1_1.loop();
//Vidéo 1_2
movie1_2 = new Movie(this, «1_2.mp4»);
movie1_2.loop();
//Vidéo 1_3
```

```
movie1_3 = new Movie(this, «1_3.mp4»);
movie1_3.loop();
//Vidéo 1_4
movie1_4 = new Movie(this, «1_4.mp4»);
movie1_4.loop();
//Bouton interaction 1
image1 = loadImage(«bouton_interaction.png»);
//charger une image, ici l'image du bouton interaction //Bouton
interaction 2
image2 = loadImage(«bouton_interaction.png»);
//Bouton interaction 3
image3 = loadImage(«bouton_interaction.png»);
//Bouton interaction 4
image4 = loadImage(«bouton_interaction.png»);
} //lire les movie
void movieEvent(Movie m) {
//lire la vidéo m.read();}
void draw(){
//Apparition de la vidéo 1
image(movie1, 0, 0, width, height);
//apparition de la vidéo nommée movie1, de la taille de l'écran //SON
if(millis() > 0){
//si le temps est inférieur à 0 seconde song.play();}
//lancer le son //Timer 1
if(millis() > 21000){ //Lancement après 4s
image(image1, x_b1, y_b1, 2*r_b1, 2*r_b1);
// faire apparaître l'image 1, de position x_b1 et y_b1 indiquer plus haut
dans le code, d'une taille de deux fois le rayon indiqué plus haut //
zone de click sur le bouton :
//x_b = position en x du bouton
//y_b = position en y du bouton
//2*r_b = diamètre du cercle (r = rayon)
fill(255);
//couleur //Action sur bouton1
if(etatBouton1){// si le bouton est pressé
//println(«The mouse is Clicked»);
etatBouton1 = true; // mémorisation de l'état «appuyé» du bouton
image(movie1_1,0,0,width,height);} } //apparition de la vidéo 1_1, sa
position et sa taille
```



```
//do stuff
//Timer 2
if(millis() > 49500){ //Lancement après 4s
image(image2, x_b2, y_b2, 2*r_b2, 2*r_b2);
//zone de click sur le bouton :
//x_b = position en x du bouton
//y_b = position en y du bouton
//2*r_b = diamètre du cercle (r = rayon)
fill(255);
//Action sur bouton2
if(etatBouton2){ // si le bouton est pressé
//println(«The mouse is Clicked»);
etatBouton2 = true; // mémorisation de l'état «appuyé» du bouton
image(movie1_2,0,0,width,height);} } //apparition de la vidéo 1_2, sa
position et sa taille
//do stuff
//Timer 3
if(millis() > 48800){ //Lancement après 4s
image(image3, x_b3, y_b3, 2*r_b3, 2*r_b3);
//zone de click sur le bouton :
//x_b = position en x du bouton
//y_b = position en y du bouton
//2*r_b = diamètre du cercle (r = rayon)
fill(255);
//Action sur bouton3
if(etatBouton3){ // si le bouton est pressé
//println(«The mouse is Clicked»);
etatBouton3 = true; // mémorisation de l'état «appuyé» du bouton
image(movie1_3,0,0,width,height);} } //apparition de la vidéo 1_3, sa
position et sa taille
//do stuff
//Timer 4
if(millis() > 52800){ //Lancement après 4s
image(image4, x_b4, y_b4, 2*r_b4, 2*r_b4);
//zone de click sur le bouton :
//x_b = position en x du bouton
//y_b = position en y du bouton
//2*r_b = diamètre du cercle (r = rayon)
fill(255);
```

```
//Action sur bouton4
if(etatBouton4){ // si le bouton est pressé
//println(«The mouse is Clicked»);
etatBouton4 = true; // mémorisation de l'état «appuyé» du bouton
image(movie1_4,0,0,width,height);}} //apparition de la vidéo 1_4, sa
position et sa taille
//do stuff
//Zone de click sur le bouton1
void mousePressed(){
//lorsque l'on click sur le bouton if ( (sq(mouseX-(x_b1+r_
b1))+sq(mouseY-(y_b1+r_b1)))<=sq(r_b1)){
//équation pour que la zone de click se situe sur le bouton
//sq() = au carré
etatBouton1 = !etatBouton1; // mémorisation de l'état «appuyé» du
bouton
} //Zone de click sur le bouton2
if ( (sq(mouseX-(x_b2+r_b2))+sq(mouseY-(y_b2+r_b2)))<=sq(r_b2)){
//équation pour que la zone de click se situe sur le bouton
//sq() = au carré
etatBouton2 = !etatBouton2; // mémorisation de l'état «appuyé» du
bouton
} //Zone de click sur le bouton3
if ( (sq(mouseX-(x_b3+r_b3))+sq(mouseY-(y_b3+r_b3)))<=sq(r_b3)){
//équation pour que la zone de click se situe sur le bouton
//sq() = au carré
etatBouton3 = !etatBouton3;} // mémorisation de l'état «appuyé» du
bouton
//Zone de click sur le bouton4
if ( (sq(mouseX-(x_b4+r_b4))+sq(mouseY-(y_b4+r_b4)))<=sq(r_b4))
//équation pour que la zone de click se situe sur le bouton
//sq() = au carré
{ etatBouton4 = !etatBouton4; }} // mémorisation de l'état «appuyé»
du bouton
void stop() {
minim.stop();
//stopper le son super.stop();}
```

Suite à ce projet d'OpenFrac#2, j'ai la possibilité de retravailler avec le FRAC PACA. Effectivement, ils réalisent de nombreux outils pédagogiques afin d'interagir avec leur public et de les sensibiliser à l'Art contemporain. On peut se référer à l'interview d'Annabelle Arnaud pour en savoir plus. Leur prochaine réalisation consiste à créer une valise pédagogique présentant le travail de l'artiste Olivier Rebufa. Le FRAC m'a donc proposé de réaliser une vidéo interactive qui sera contenu dans la valise pédagogique et qui permettra de découvrir comment l'artiste crée ses photographies. En complément, je vais devoir mettre en page, sous forme d'édition papier et numérique, une série de documents présentant le travail d'Olivier Rebufa. Il serait donc intéressant de voir comment tous ses supports peuvent s'articuler entre eux.

