M1 ENJMIN - MJV102

Conception et développement informatique

2 à 6 oct. 2017

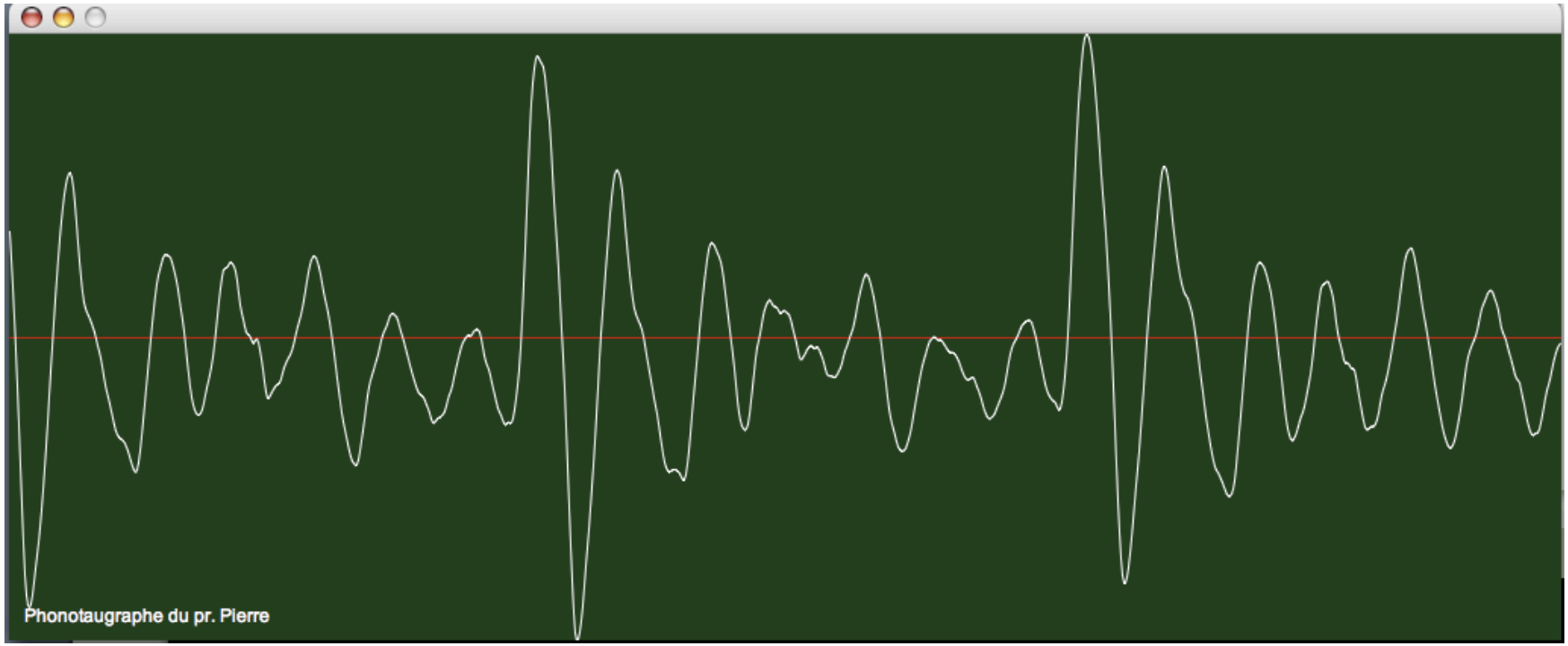
Pierre Cubaud cubaud @ cnam.fr
Viviane Gal gal @ cnam.fr

le cnam
École nationale du jeu et des médias interactifs numériques enjmin

5. Captation

Le son avec la librairie Minim

Exemple d'un oscilloscope

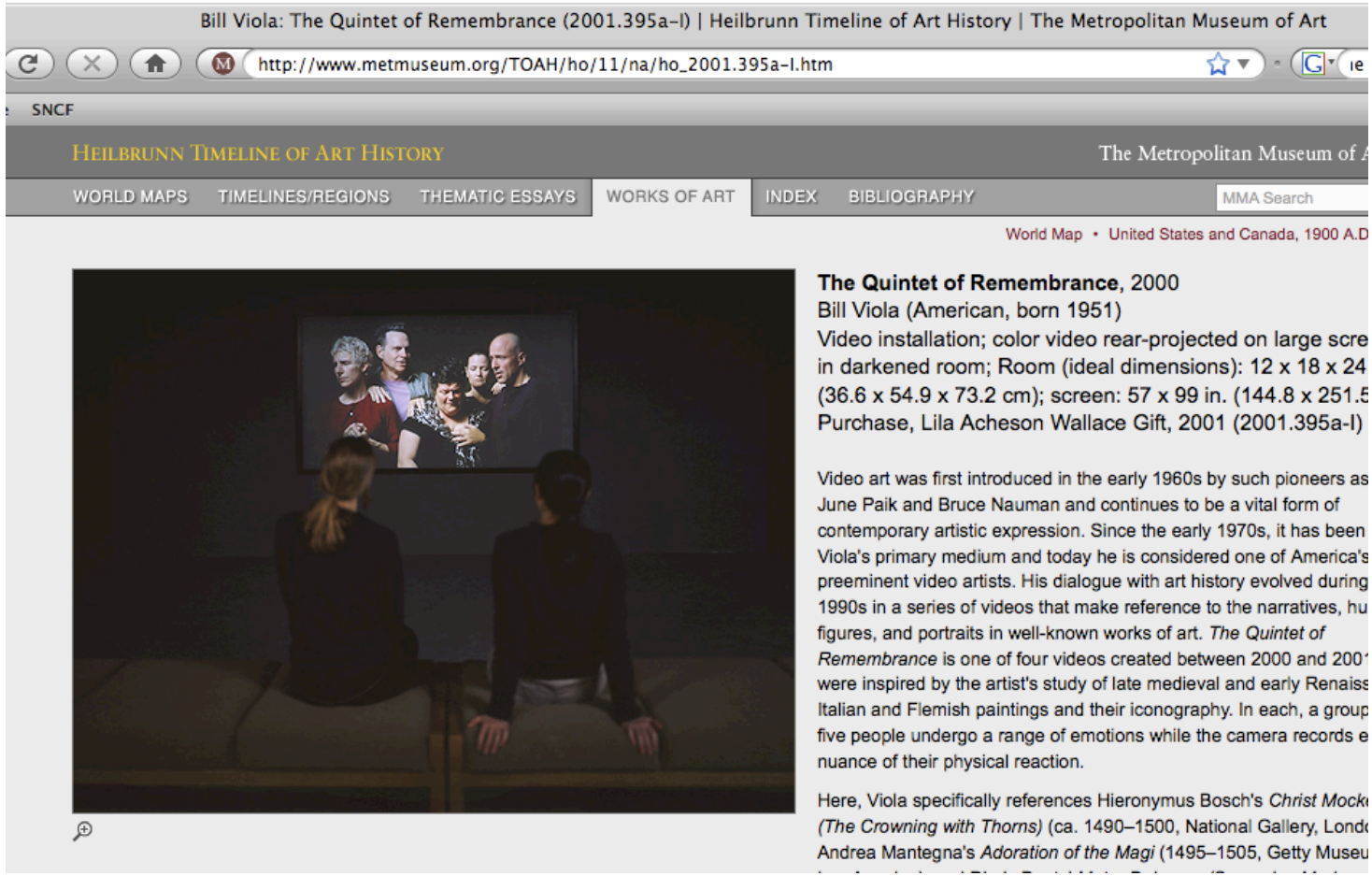


```
import ddf.minim.*;

Minim minim;
AudioInput in;
float lemin, lemax;
float[] X = new float[1024];

void setup(){
  size(800, 600);
  minim = new Minim(this);
  in = minim.getLineIn
        (Minim.MONO, 1024, 44100);
}
```

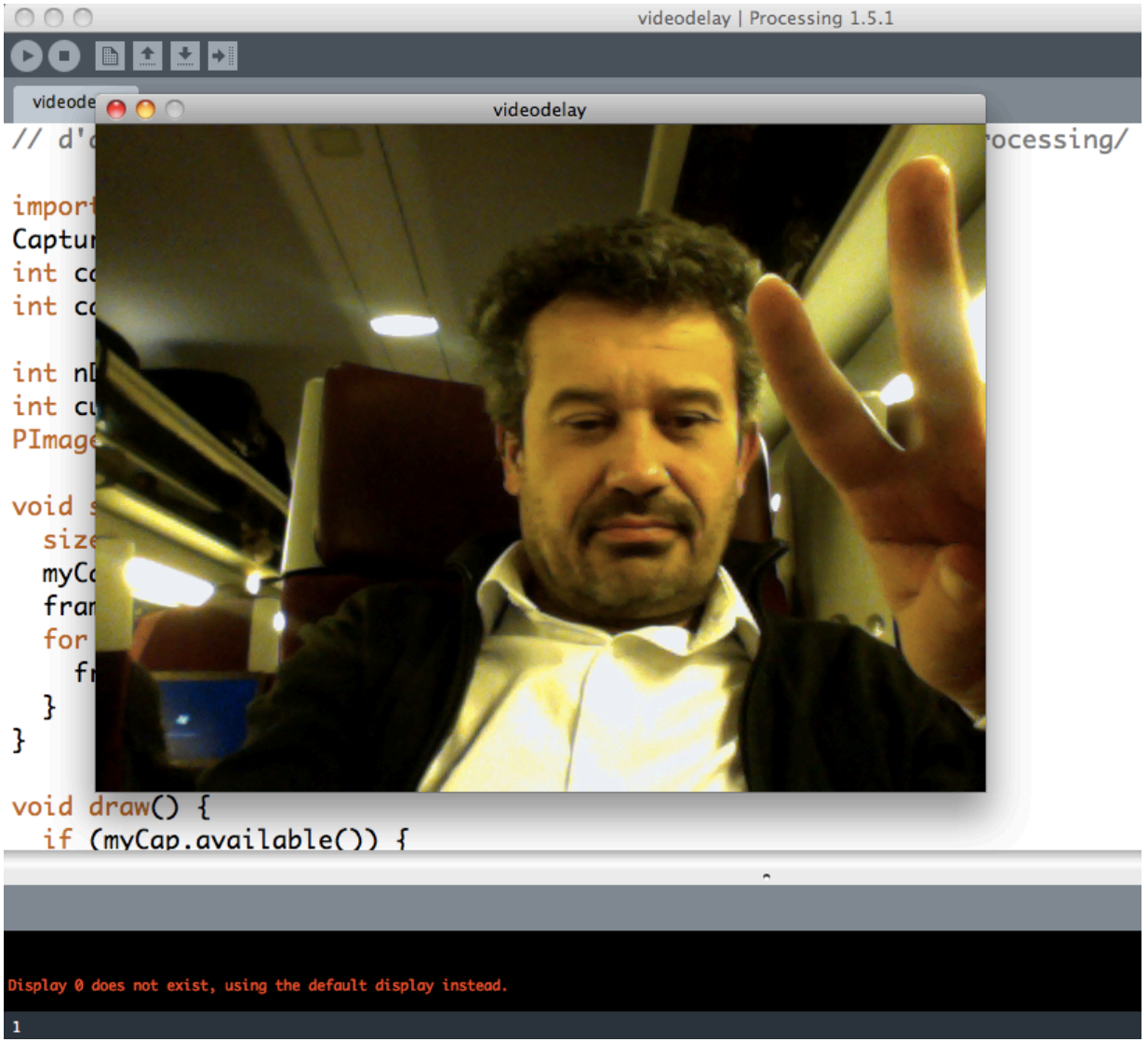
```
void draw() {
  background(15,50,20);
  fill(255);
  line(0,height/2,width,height/2);
  stroke(255);
  X = in.left.toArray();
  for(int i = 0; i < X.length-1; i++)  {
    float h1 = map(X[i],-1,+1,height,0);
    float h2 = map(X[i+1],-1,+1,height,0);
    line(i, h1, i+1, h2);
  }
}

void stop() {
  in.close();
  minim.stop();
  super.stop();
}
```

(monscope.pde)

Bill Viola


Bill Viola: The Quintet of Remembrance (2001.395a–l) | Heilbrunn Timeline of Art History | The Metropolitan Museum of Art
http://www.metmuseum.org/TOAH/ho/11/na/ho_2001.395a-l.htm
SNCF
HEILBRUNN TIMELINE OF ART HISTORY
The Metropolitan Museum of
WORLD MAPS
TIMELINES/REGIONS
THEMATIC ESSAYS
WORKS OF ART
INDEX
BIBLIOGRAPHY
MMA Search
World Map • United States and Canada, 1900 A.D
The Quintet of Remembrance, 2000
Bill Viola (American, born 1951)
Video installation; color video rear-projected on large scre
in darkened room; Room (ideal dimensions): 12 x 18 x 24
(36.6 x 54.9 x 73.2 cm); screen: 57 x 99 in. (144.8 x 251.5
Purchase, Lila Acheson Wallace Gift, 2001 (2001.395a-l)
Video art was first introduced in the early 1960s by such pioneers as
June Paik and Bruce Nauman and continues to be a vital form of
contemporary artistic expression. Since the early 1970s, it has been
Viola's primary medium and today he is considered one of America's
preeminent video artists. His dialogue with art history evolved during
1990s in a series of videos that make reference to the narratives, hu
figures, and portraits in well-known works of art. The Quintet of
Remembrance is one of four videos created between 2000 and 200
were inspired by the artist's study of late medieval and early Renaiss
Italian and Flemish paintings and their iconography. In each, a group
five people undergo a range of emotions while the camera records e
nuance of their physical reaction.
Here, Viola specifically references Hieronymus Bosch's Christ Mock
(The Crowning with Thorns) (ca. 1490–1500, National Gallery, Lond
Andrea Mantegna's Adoration of the Magi (1495–1505, Getty Museu

demo videodelay



```
// d'apres http://www.flong.com/blog/2011/free-video-delay-in-processing/

import processing.video.*;
Capture myCap;
int capW = 640; //320;
int capH = 480; //240;

int nDelayFrames = 50; //100; // about 3 seconds
int currentFrame = nDelayFrames-1;
PImage frames[];

void setup() {
  size (640, 480);
  myCap = new Capture(this, capW, capH);
  frames = new PImage[nDelayFrames];
  for (int i=0; i<nDelayFrames; i++) {
    frames[i] = createImage(capW, capH, ARGB);
  }
}

void draw() {
  if (myCap.available()) {
    myCap.read();
    frames[currentFrame].loadPixels();
    arrayCopy (myCap.pixels, frames[currentFrame].pixels);
    frames[currentFrame].updatePixels();
    currentFrame = (currentFrame-1 + nDelayFrames) % nDelayFrames;
  }
  image (frames[currentFrame], 0, 0, width, height);
}
```

capture de l'écran

capture

```
import processing.video.*;
MovieMaker mm;  // Declare MovieMaker object

void setup() {
  size(600, 600);
  mm = new MovieMaker(this, 600, 600, "resultat.mov");
}

void draw() {
  ellipse(mouseX,mouseY,50,50);
  mm.addFrame();  // Add window's pixels to movie
}

void keyPressed() {
  mm.finish();  // Finish the movie
}
```

Tracking video (vraiment) minimal

Préalable :

se fabriquer une cible

mais
pourquoi en rouge ?

trackVideo

```
import processing.video.*;
Capture video;

void setup(){
  size(640, 480);
  video = new Capture(this, width, height, 30);
  video.start(); // pour Processing 2.0 et +
  noStroke();
  smooth();
}
void draw() {
if (video.available()){
  float hmax = 0;
  int tx = 0;
  int ty = 0;
  video.read();
  video.loadPixels();
  int index = 0;
  for (int y = 0; y < video.height; y++) {
   for (int x = 0; x < video.width; x++) {
    float h = hue(video.pixels[index]);
    // la teinte max donne du rouge
    if (h > hmax){
     hmax = h; tx = x; ty = y;
    }
    index++;
   }
  }
 // feedback graphique
 image(video, 0, 0, width, height);
 fill(255, 255, 0, 128);
 ellipse(tx, ty, 100, 100);
}
}
```

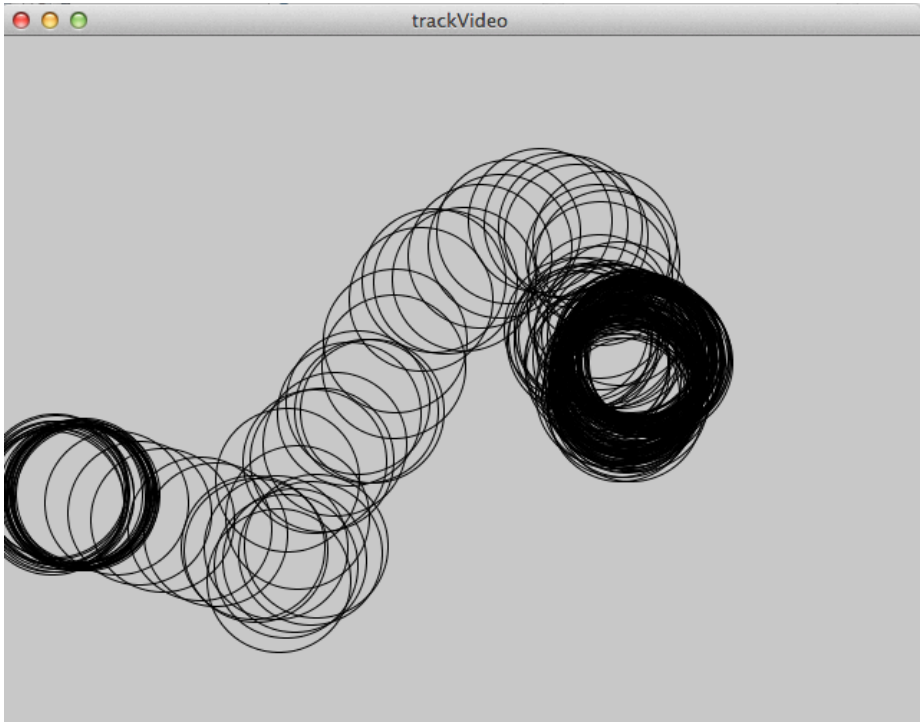
- étudier l'impact de l'appel video.available()

- enlever image() et transformer le dessin en un curseur qui évolue dans le sens du mouvement : comme un miroir

```
import processing.video.*;
Capture video;
int tx = 0;
int ty = 0;
float lemax = 0;

void setup() {
  size(640, 480);
  video = new Capture(this, width, height, 30);
  video.start();
  noStroke();
  smooth();
}
```

```
void draw() {
  if (video.available()) {
    video.read();
    video.loadPixels();
    int index = 0;
    for (int y = 0; y < video.height; y++) {
      for (int x = 0; x < video.width; x++){
        float h = hue(video.pixels[index]);
        // la teinte max donne du rouge
        if (h > lemax){
          lemax = h;
          tx = x;
          ty = y;
        }
        index++;
      }
    }
  // feedback graphique
    image(video, 0, 0, width, height);
    fill(255, 255, 0, 128);
    ellipse(tx, ty, 100, 100);
  }
}
```

Utilisable comme curseur pour dessiner ?



problème de bruit ! (tremblements, aléa de l'analyse)

Une solution pour "retirer" le bruit de captation :
utiliser un filtre passe-bas

variante simple très utilisée : le filtre alpha

$$y_{n+1} = (1-\alpha).y_n + \alpha.x_{n+1}$$

nouvelle estimation — **ancienne estimation** — **nouvelle mesure**

si alpha fort (>0.50) : réponse rapide mais bruitée
sinon : moins de bruit mais inertie

=> il faut tester (ici alpha = 30% = 0.3 pas mal)

modification du code :

```
import processing.video.*;
Capture video;
float txe = 0;
float tye = 0;
float alpha = 0.30;
```

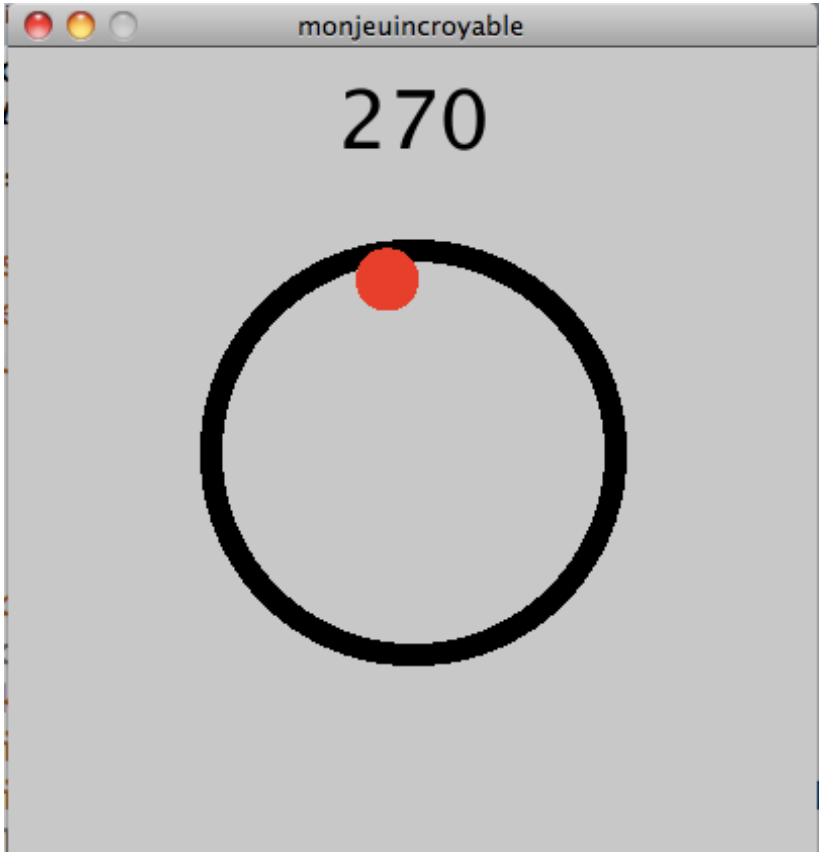
nouvelles déclarations au début du code

```
  }
 // feedback graphique
 image(video, 0, 0, width, height);
 txe = (1-alpha)*txe + alpha*tx;
 tye = (1-alpha)*tye + alpha*ty;
 stroke(255,255,0);
 ellipse(txe, tye, 100, 100);
}
}
```

à la fin, filtrage avant le tracé

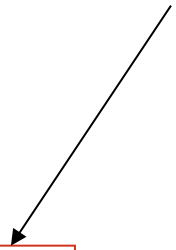
- tester avec des alpha forts et faibles
- gestion de l'echec de tracking ????

Mon jeu incroyable (v1)




```
score = 1000;
while (curseur_pas_sur_la cible()) {
  score--;
  if (score<0) noLoop();
}
```

```
int score=300;
int RAYON = 100;
int x,y;

void setup(){
  size(400,400);
  noCursor();
  x = width/2;
  y = height/2-100;
}

void draw(){
  // objet guide
  background(200);
  noFill();stroke(0);strokeWeight(10);
  ellipse(width/2,height/2,2*RAYON,2*RAYON);
  // le pointeur
  x = mouseX;
  y = mouseY;
  fill(250,0,0);noStroke();
  ellipse(x+random(-10,10),y+random(-10,10),30,30);
  // le score
  float d = sqrt((x-width/2)*(x-width/2)+(y-height/2)*(y-height/2));
  if ((d < RAYON-5)||(d > RAYON+5)) score--;
  fill(0);textSize(40);textAlign(CENTER);text(str(score),width/2,50);
  // arret du jeu
  if (score==0) noLoop();
}
```



pour
simuler
le capteur

monjeuVIDEO

```
int score=30000;
int RAYON = 100;
int x,y;
import processing.video.*;
Capture video;
float txe = 0;
float tye = 0;
float alpha = 0.10;

void setup(){
  size(400,400);
  noCursor();
  video = new Capture(this, width, height, 30);
  //video.start(); // pour Processing 2.0 et +
  x = width/2;
  y = height/2-100;
}
```

```
void draw(){
  background(200);
  //image(video, 0, 0, width, height);
  // objet guide
  noFill();stroke(0);strokeWeight(10);
  ellipse(width/2,height/2,2*RAYON,2*RAYON);
  // le pointeur
  curseurVideo();
  x = width-(int)txe;
  y = (int)tye;
  fill(250,0,0);noStroke();
  ellipse(x,y,30,30);
  // le score
  float d = sqrt((x-width/2)*(x-width/2)+(y-height/2)*(y-height/2));
  if ((d < RAYON-5)||(d > RAYON+5)) score--;
  fill(0);textSize(40);textAlign(CENTER);text(str(score),width/2,50);
  // arret du jeu
  if (score==0) noLoop();
}
```

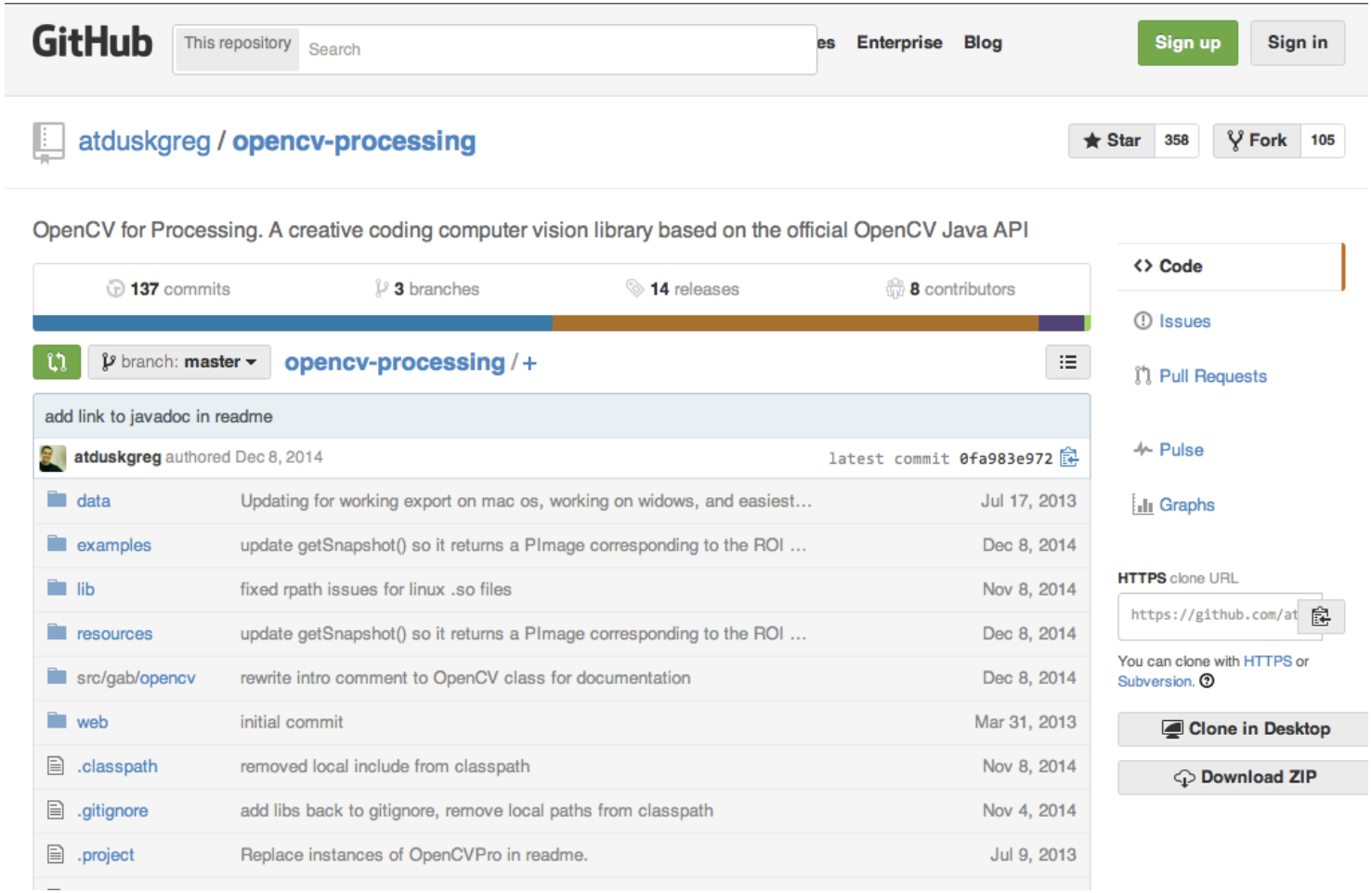
```
void curseurVideo(){
 float hmax = 0;
 int tx = 0;
 int ty = 0;
 video.read();
 video.loadPixels();
 int index = 0;
 for (int y = 0; y < video.height; y++) {
   for (int x = 0; x < video.width; x++) {
    float h = hue(video.pixels[index]);
    // la teinte max donne du rouge
    if (h > hmax){
     hmax = h; tx = x; ty = y;
    }
    index++;
   }
 }
 txe = (1-alpha)*txe + alpha*tx;
 tye = (1-alpha)*tye + alpha*ty;
}
```

opencv.org

librairie créée chez Intel en Russie en 1999-2000

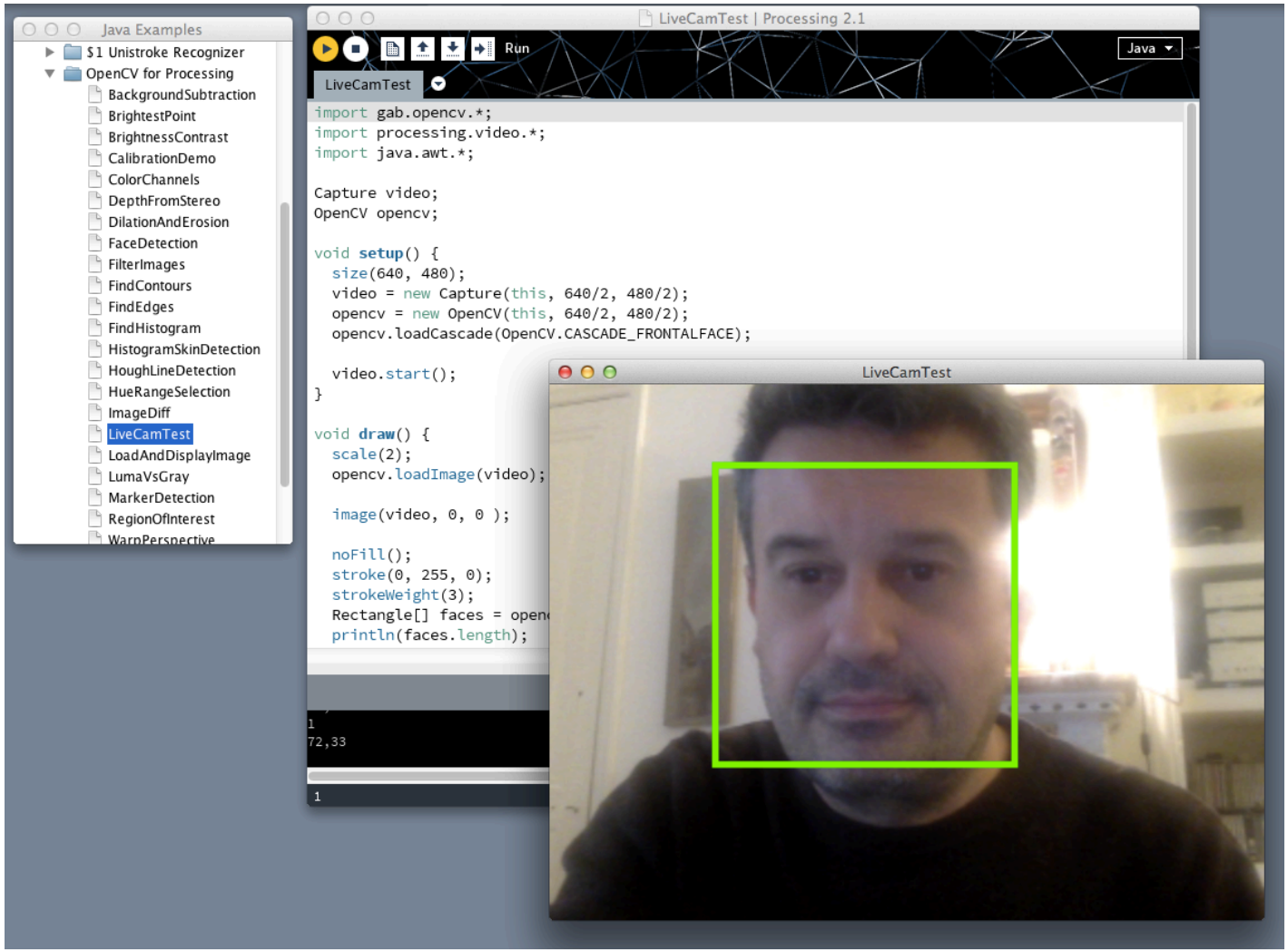
installation pour Processing

https://github.com/atduskgreg/opencv-processing



rque : il existe aussi un portage plus ancien fait aux Beaux-Arts d'Aix

Test de la reconnaissance faciale (demo "LiveCamTest")



Viola & Jones, 2001-4

Méthode de Viola et Jones

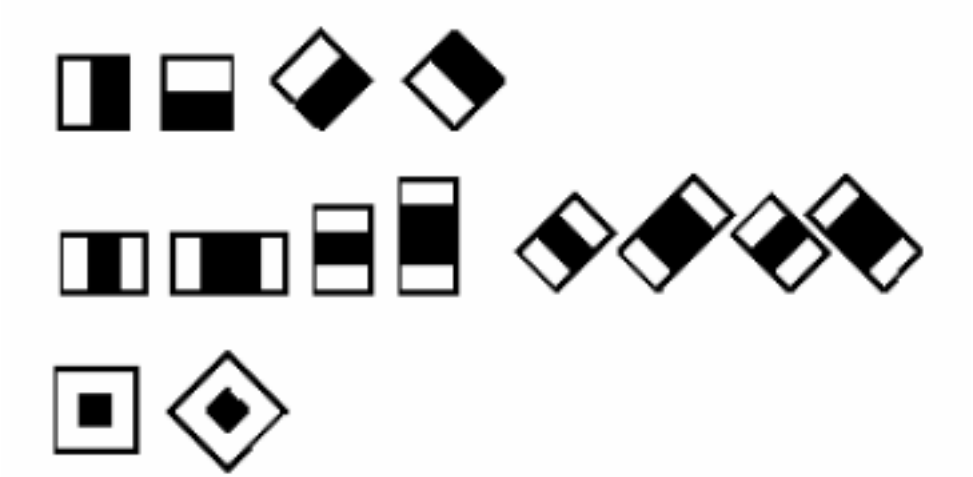
Vous lisez un « article de qualité ».

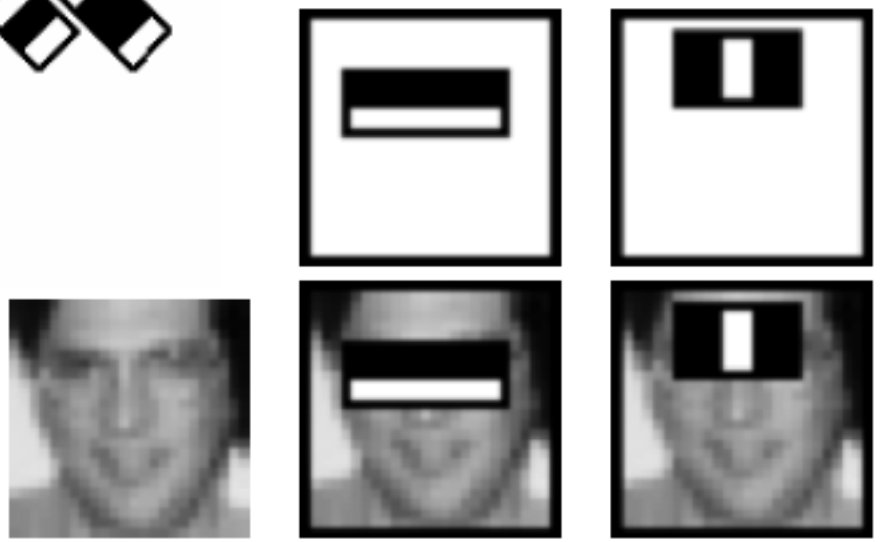
Un exemple de détection de visage par la méthode de Viola et Jones.

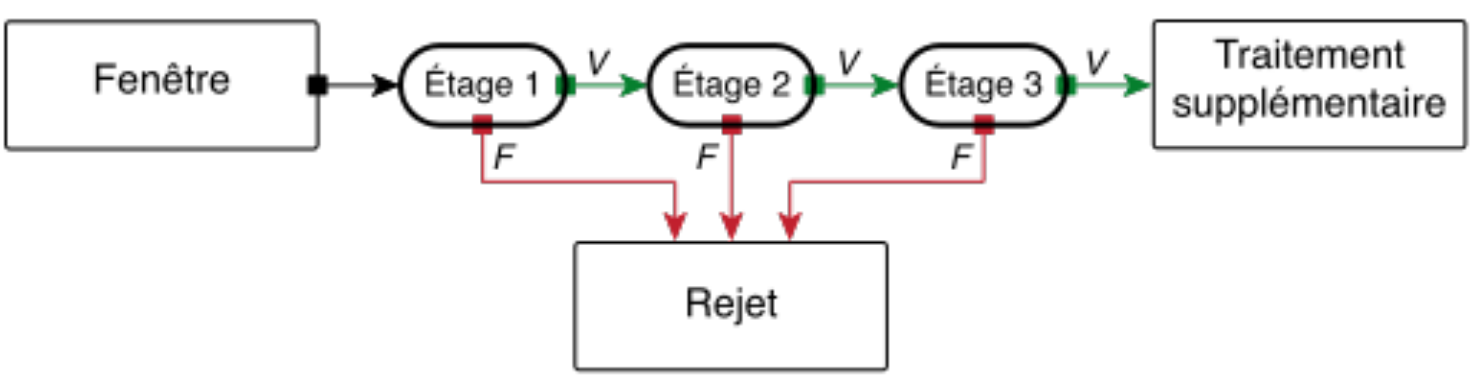
La **méthode de Viola et Jones** est une méthode de détection d'objet dans une image numérique, proposée par les chercheurs Paul Viola et Michael Jones en 2001. Elle fait partie des toutes premières méthodes capables de détecter efficacement et en temps réel des objets dans une image. Inventée à l'origine pour détecter des visages, elle peut également être utilisée pour détecter d'autres types d'objets comme des voitures ou des avions. La méthode de Viola et Jones est l'une des méthodes les plus connues et les plus utilisées, en particulier pour la détection de visages et la détection de personnes.

En tant que procédé d'apprentissage supervisé, la méthode de Viola et Jones nécessite de quelques centaines à plusieurs milliers d'exemples de l'objet que l'on souhaite détecter, pour entraîner un classifieur. Une fois son apprentissage réalisé, ce classifieur est utilisé pour détecter la présence éventuelle de l'objet dans une image en parcourant celle-ci de manière exhaustive, à toutes les positions et dans toutes les tailles possibles.

Considérée comme étant l'une des plus importantes méthodes de détection d'objet, la méthode de Viola et Jones est notamment connue pour avoir introduit plusieurs notions reprises ensuite par de nombreux chercheurs en vision par ordinateur, à l'exemple de la notion d'image intégrale ou de la méthode de classification construite






Fenêtre
Étage 1
V
Étage 2
V
Étage 3
V
Traitement supplémentaire
F
F
F
Rejet

Une variante …

face2shape | Processing 2.1

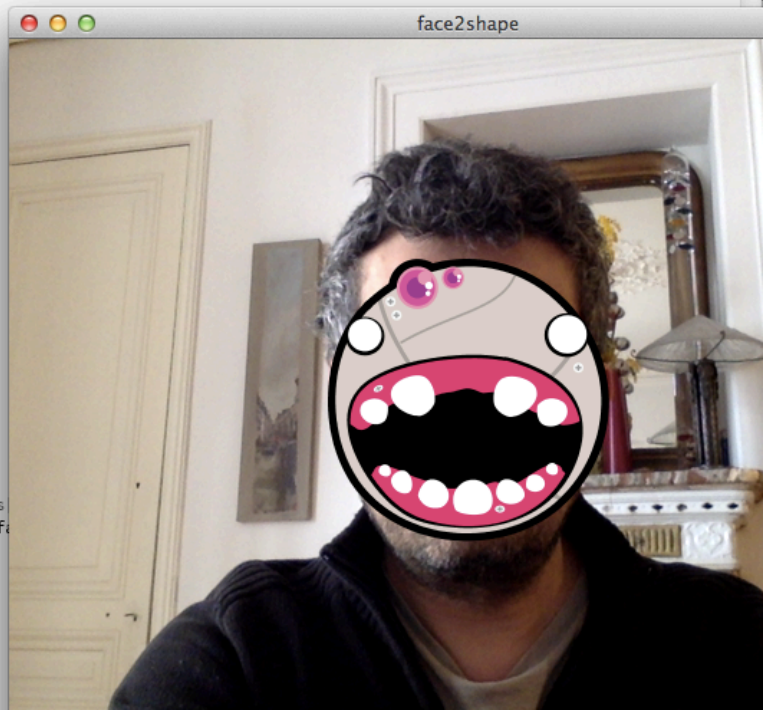
Run

Java

face2shape

```
import gab.opencv.*;
import processing.video.*;
import java.awt.*;
Capture video;
OpenCV opencv;
PShape bot;

void setup() {
  size(640, 480);
  video = new Capture(this, 640, 480);
  opencv = new OpenCV(this, 640, 480);
  opencv.loadCascade(OpenCV.CASCADE_FRONTALFACE);
  bot = loadShape("bot1.svg");

  video.start();
}

void draw() {
  //scale(2);
  opencv.loadImage(video);

  image(video, 0, 0 );

  noFill();
  stroke(0, 255, 0);
  strokeWeight(3);
  Rectangle[] faces = opencv.detect();
//  println(faces.length);

  for (int i = 0; i < faces.length; i++) {
//    println(faces[i].x + "," + faces[i].y);
    //rect(faces[i].x, faces[i].y, faces[i].width, faces
    shape(bot, faces[i].x, faces[i].y, faces[i].width, fa
  }
}

void captureEvent(Capture c) {
  c.read();
}
```

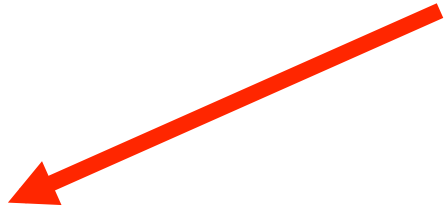
face2shape



marche aussi avec une image fixe

FaceDetectionFIXE

```
import gab.opencv.*;
import java.awt.Rectangle;

OpenCV opencv;
Rectangle[] faces;

void setup() {
  opencv = new OpenCV(this, "test.jpg");
  size(opencv.width, opencv.height);
  opencv.loadCascade(OpenCV.CASCADE_FRONTALFACE);
  faces = opencv.detect();
}

void draw() {
  image(opencv.getInput(), 0, 0);
  noFill(); stroke(0, 255, 0); strokeWeight(3);
  for (int i = 0; i < faces.length; i++) {
    rect(faces[i].x, faces[i].y, faces[i].width, faces[i].height);
  }
}
```



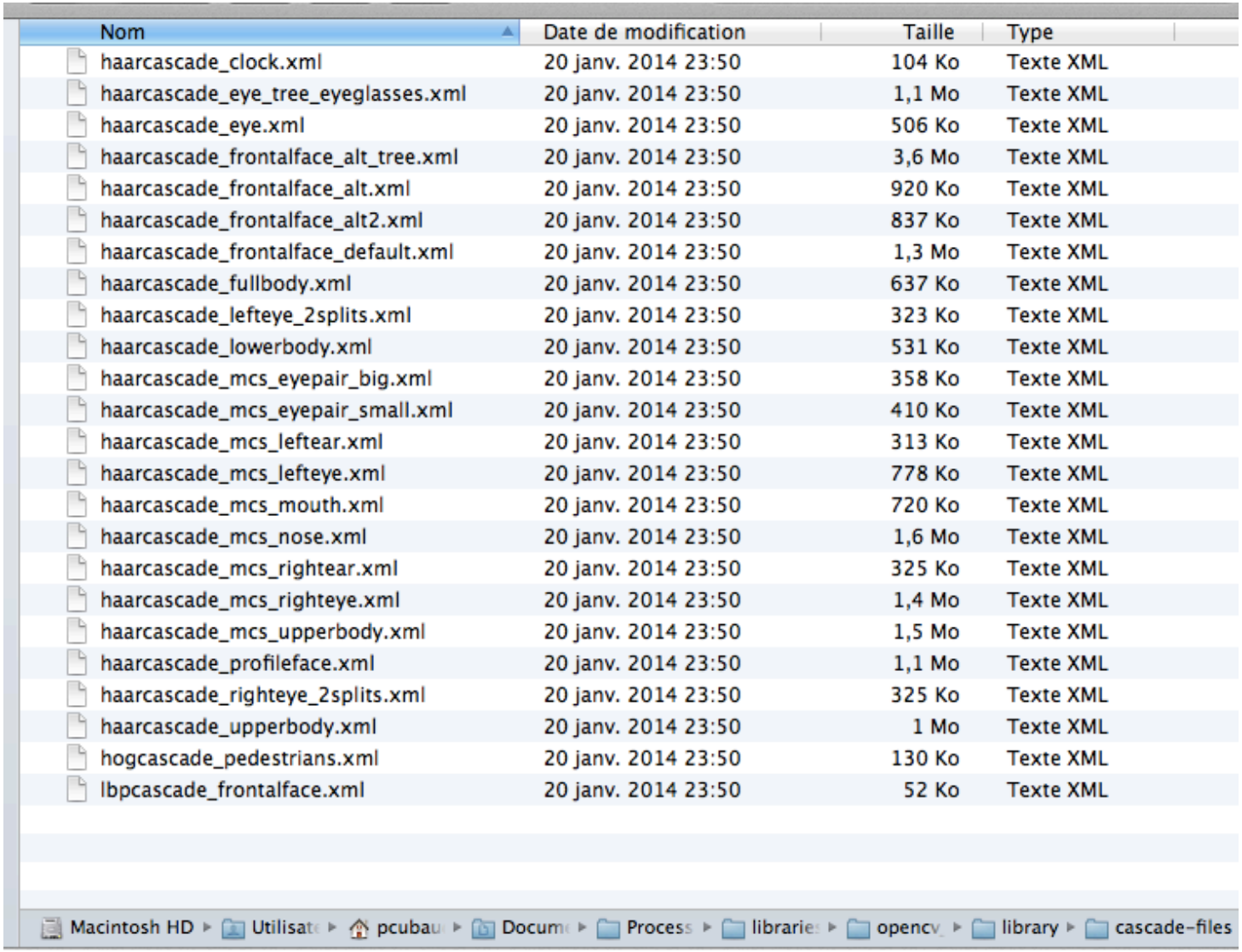
Variante pour les yeux

EyeDetectionCAM

```
Capture video;
OpenCV opencv;

void setup() {
  size(640, 480);
  video = new Capture(this, 640, 480);
  opencv = new OpenCV(this, 640, 480);
  opencv.loadCascade(OpenCV.CASCADE_EYE);
  video.start();
}

void draw() {
  image(video, 0, 0 );
  opencv.loadImage(video);
  Rectangle[] faces = opencv.detect();
  noFill();  stroke(0, 255, 0);  strokeWeight(3);
  for (int i = 0; i < faces.length; i++) {
    rect(faces[i].x, faces[i].y, faces[i].width, faces[i].height);
  }
}

void captureEvent(Capture c) {
```

Seule chose qui change :-)

Autres cascades à tester ...



Nom	Date de modification	Taille	Type
haarcascade_clock.xml	20 janv. 2014 23:50	104 Ko	Texte XML
haarcascade_eye_tree_eyeglasses.xml	20 janv. 2014 23:50	1,1 Mo	Texte XML
haarcascade_eye.xml	20 janv. 2014 23:50	506 Ko	Texte XML
haarcascade_frontalface_alt_tree.xml	20 janv. 2014 23:50	3,6 Mo	Texte XML
haarcascade_frontalface_alt.xml	20 janv. 2014 23:50	920 Ko	Texte XML
haarcascade_frontalface_alt2.xml	20 janv. 2014 23:50	837 Ko	Texte XML
haarcascade_frontalface_default.xml	20 janv. 2014 23:50	1,3 Mo	Texte XML
haarcascade_fullbody.xml	20 janv. 2014 23:50	637 Ko	Texte XML
haarcascade_lefteye_2splits.xml	20 janv. 2014 23:50	323 Ko	Texte XML
haarcascade_lowerbody.xml	20 janv. 2014 23:50	531 Ko	Texte XML
haarcascade_mcs_eyepair_big.xml	20 janv. 2014 23:50	358 Ko	Texte XML
haarcascade_mcs_eyepair_small.xml	20 janv. 2014 23:50	410 Ko	Texte XML
haarcascade_mcs_leftear.xml	20 janv. 2014 23:50	313 Ko	Texte XML
haarcascade_mcs_lefteye.xml	20 janv. 2014 23:50	778 Ko	Texte XML
haarcascade_mcs_mouth.xml	20 janv. 2014 23:50	720 Ko	Texte XML
haarcascade_mcs_nose.xml	20 janv. 2014 23:50	1,6 Mo	Texte XML
haarcascade_mcs_rightear.xml	20 janv. 2014 23:50	325 Ko	Texte XML
haarcascade_mcs_righteye.xml	20 janv. 2014 23:50	1,4 Mo	Texte XML
haarcascade_mcs_upperbody.xml	20 janv. 2014 23:50	1,5 Mo	Texte XML
haarcascade_profileface.xml	20 janv. 2014 23:50	1,1 Mo	Texte XML
haarcascade_righteye_2splits.xml	20 janv. 2014 23:50	325 Ko	Texte XML
haarcascade_upperbody.xml	20 janv. 2014 23:50	1 Mo	Texte XML
hogcascade_pedestrians.xml	20 janv. 2014 23:50	130 Ko	Texte XML
lbpcascade_frontalface.xml	20 janv. 2014 23:50	52 Ko	Texte XML

Macintosh HD ▸ Utilisat ▸ pcubau ▸ Docum ▸ Process ▸ librarie ▸ opencv ▸ library ▸ cascade-files

Reprise de mon jeu incroyable

C'est trop fatiguant avec sa vraie tête