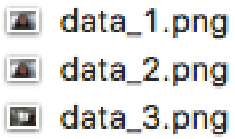
Plz take a cloz look

A BRIEF HISTORY OF CASUAL TROLLING

CONCEPT

The screen displays one first image inviting the viewer to «have a close look»

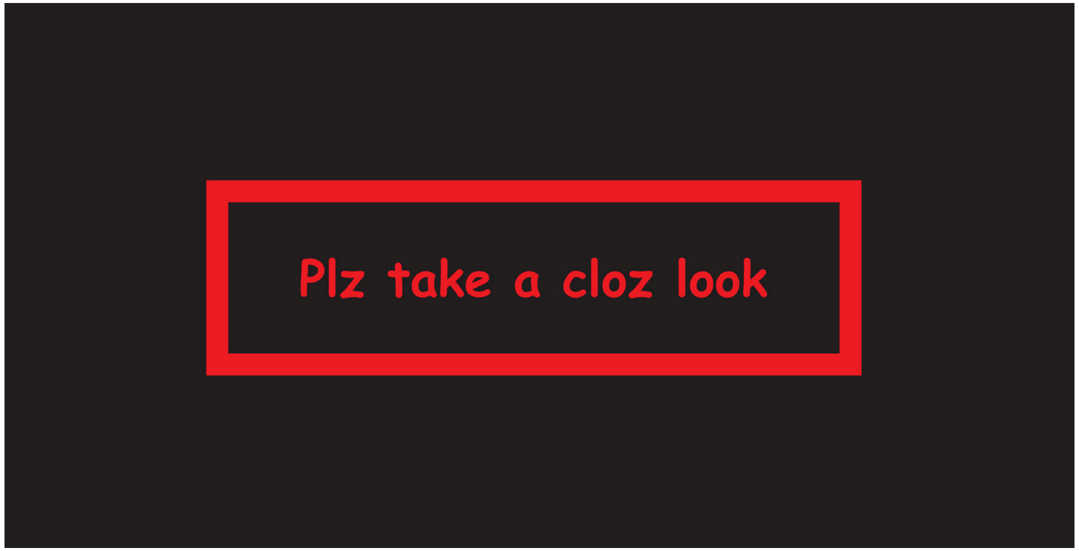
Through face recognition (which the viewer in unaware of), the programm starts taking screen-shots of the camera while the viewer is in the frame.



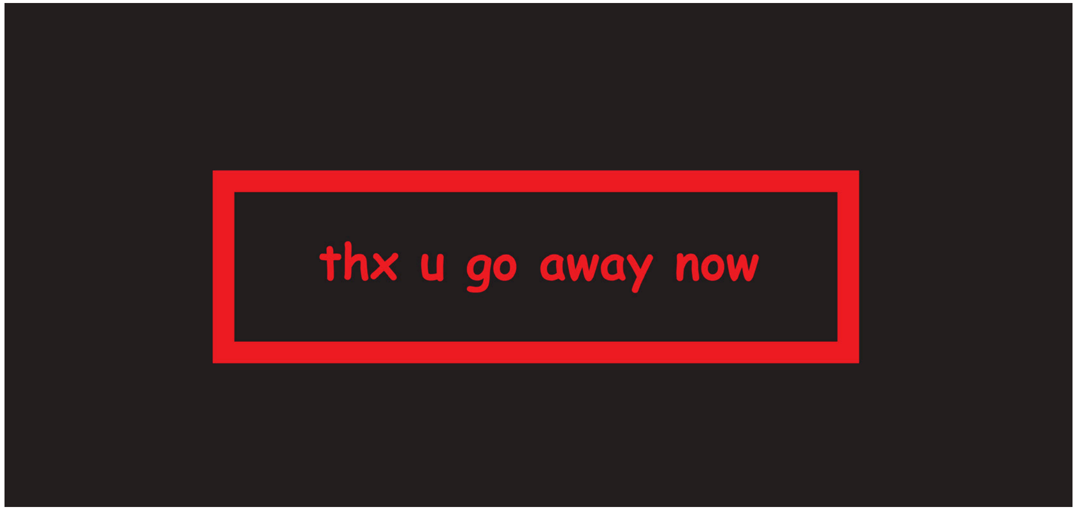
data_1.png
data_2.png
data_3.png

Data generated by the code

Once the viewer is detected, the screen displays the second message, «thanks, you (can) go away now», implying that the computer has noticed the person's presence.



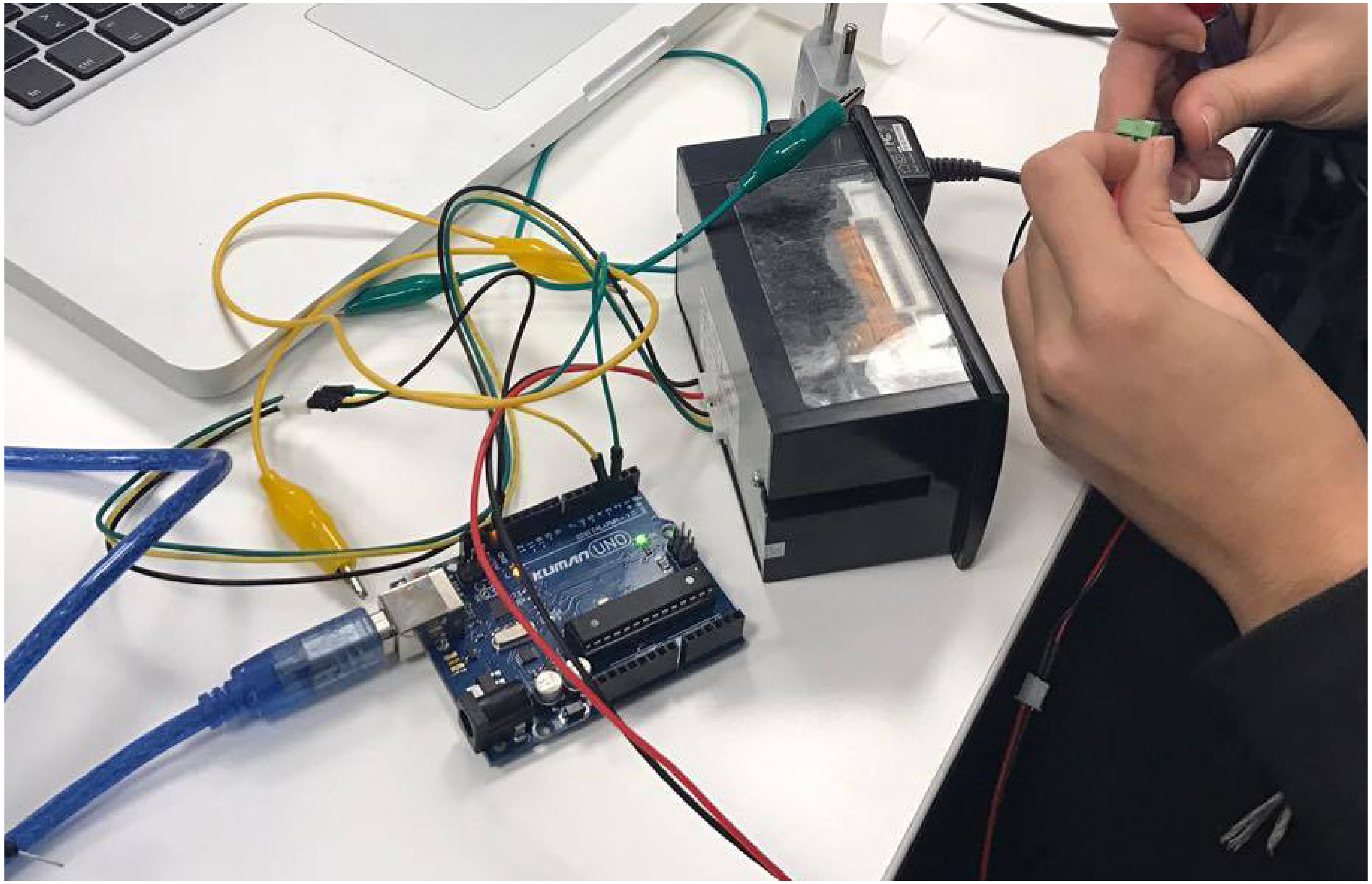
Plz take a cloz look



thx u go away now

MATERIAL

- Computer
- Internal Camera
- Processing libraries :
  - Video
  - OpenCV
  - Java.awt
- Arduino
- Thermal Printer
- Podium
- Second screen (iPad)



CODE - PROCESSING

```
import gab.opencv.*;
import processing.video.*;
import java.awt.*;
import java.util.*;
import processing.serial.*;

Serial myPort;
int nb_seconds, nb_images;

Capture video;
OpenCV opencv;
PImage b;
PImage a;

void setup() {

  fullScreen();
  String portName = Serial.list()[1];
myPort = new Serial(this, portName,
57600);
  frameRate(1);
  imageMode(CENTER);
  b = loadImage(«text2.png»);
  a = loadImage(«text1.png»);

  String[] cameras = Capture.list();
    if (cameras.length == 0) {
    println(«There are no cameras
available for capture.»);
    exit();
  } else {
    println(«Available cameras:»);}
    for (int i = 0; i < cameras.len-
gth; i++) {
      println(cameras[i]);
video = new Capture(this, 640, 480,
cameras[0]);
    video.start();
    opencv = new OpenCV(this, 640,
480);
    opencv.loadCascade(OpenCV.CASCADE_
FRONTALFACE);
  }
}

void draw() {
  myPort.write(«0\n»);
   if (video.available() == true) {
    video.read();
  }
  opencv.loadImage(video);
  background(0);
  image(a, width/2, height/2, width,
height);
 // image(video, 0, 0 );
int s = second();
int m = minute();
int h = hour();
int d = day();
int t = month();
int y = year();
  noFill();

  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[i].height);
    Date f = new Date();
    long current=f.getTime()/1000;
    nb_seconds++;
    if (nb_seconds%3 == 0) {
    nb_images++;
    video.read();
    PImage cp = video.get();
    cp.save(«../node_mosaique/
public/images/data_» + current +
«.png»);
  }   }

  if(faces.length>0) {
  image(b, width/2, height/2,
width, height);
    //text(«thanks u can go away
now», width/2, height/3);
    println(«I SEE YOU!»);
    myPort.write(y + «/» + t + «/»
+ d + «  «+ h + «:» + m + «:» + s+
«\n»);         //send a 1
    println(«1»); }
   else{
      myPort.write(«0\n»);
      println(«0»);
    }}
void captureEvent(Capture c) {
  c.read();
}
```

CODE - ARDUINO

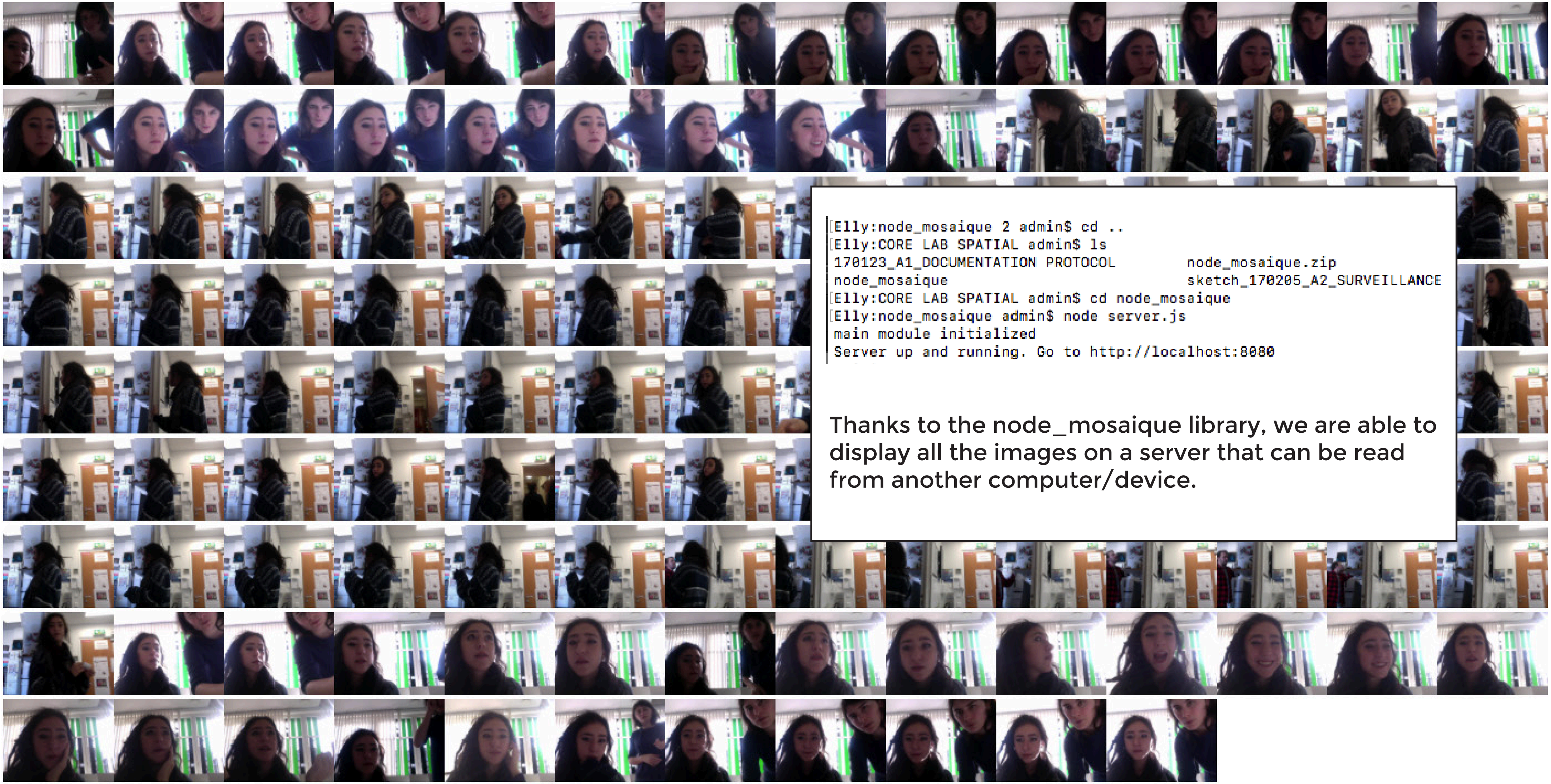
```
#include <SoftwareSerial.h>

SoftwareSerial Thermal(2, 3);
int heatTime = 80;
int heatInterval = 255;
char printDensity = 15;
char printBreakTime = 15;
String val = «0»;
void setup()
{
 Serial.begin(57600); // for debug
info to serial monitor
 Thermal.begin(19200); // to write to
our new printer
 initPrinter();
}
void initPrinter()
{
 Thermal.write(27);
 Thermal.write(55);
 Thermal.write(7);
 Thermal.write(heatTime);
 Thermal.write(heatInterval);   Ther-
mal.write(18);
 Thermal.write(35);
 int printSetting = (printDensity<<4)
| printBreakTime;
 Thermal.write(printSetting);
 Serial.println();
 Serial.println(«Printer ready»);
}
```

```
void loop()
{
  if (Serial.available())
   { // If data is available to read,
     //val = Serial.read(); // read it
and store it in val
     val = Serial.readStringUn-
til(‘\n’);
   }
   if (val != «0»)
   { // If 1 was received
     Thermal.println(val);
   }
   delay(1000); // Wait 10 millise-
conds for next reading
}
```

OUTPUT

1. DISPLAY THE SCREENSHOTS LIVE ON WEB



```
[Elly:node_mosaique 2 admin$ cd ..
[Elly:CORE LAB SPATIAL admin$ ls
 170123_A1_DOCUMENTATION PROTOCOL        node_mosaique.zip
 node_mosaique                           sketch_170205_A2_SURVEILLANCE
[Elly:CORE LAB SPATIAL admin$ cd node_mosaique
[Elly:node_mosaique admin$ node server.js
 main module initialized
 Server up and running. Go to http://localhost:8080
```

Thanks to the node_mosaique library, we are able to display all the images on a server that can be read from another computer/device.

OUTPUT

2. PRINT THE DATA

Thanks to an Arduino-connected thermalprinter, we will print - in real time - the data associated to each screenshot (date and time), creating a record of each time the device recognized a face in the room.

Technical source :
http://tronixstuff.com/2011/07/08/tutorial-arduino-and-a-thermal-printer/

```
 I see you
 I see you
you
 I see you
 I see you
 I see you
 I see you
 I see you
 I see you
 I see you
 I see you
 I see you
 I see you
 I see you
 I see you
 I see you
you
 I see you
 I see you
 I see you
 I see you
 I see you
 I see you
 I see you
 I see you
 I see you
 I see you
 I see you
 I see you
you
```