Surveillance project:
inner space

By Alina Tsyganova

Description

- Documenting a small space (i.e. living room, classroom) through detecting people’s faces and tracking their body movements through laptop’s Webcam via using OpenCV library in Processing.
- The Multiple_Color_Tracking function tracks a person and changes the color of the square frame every time a new face is detected. The set maximum number of people open for detection & tracking is 10 (the RGB color values are pre-set up in Processing).
- Every 20 seconds Processing/Webcam takes an automatic screenshot and saves it in the designated ‘Object_tracking’ folder as a jpg file in a chronological numerical order starting from 0.
- The recorded images then get automatically published on an HTML website page called ’Surveillance project’ created through Atom text editor. The webpage displays a maximum of 10 pictures.
- The page of the website is automatically refreshed every second to display new images.
- These images then get automatically printed on A4 paper through the Terminal ‘Watch’ script print command.

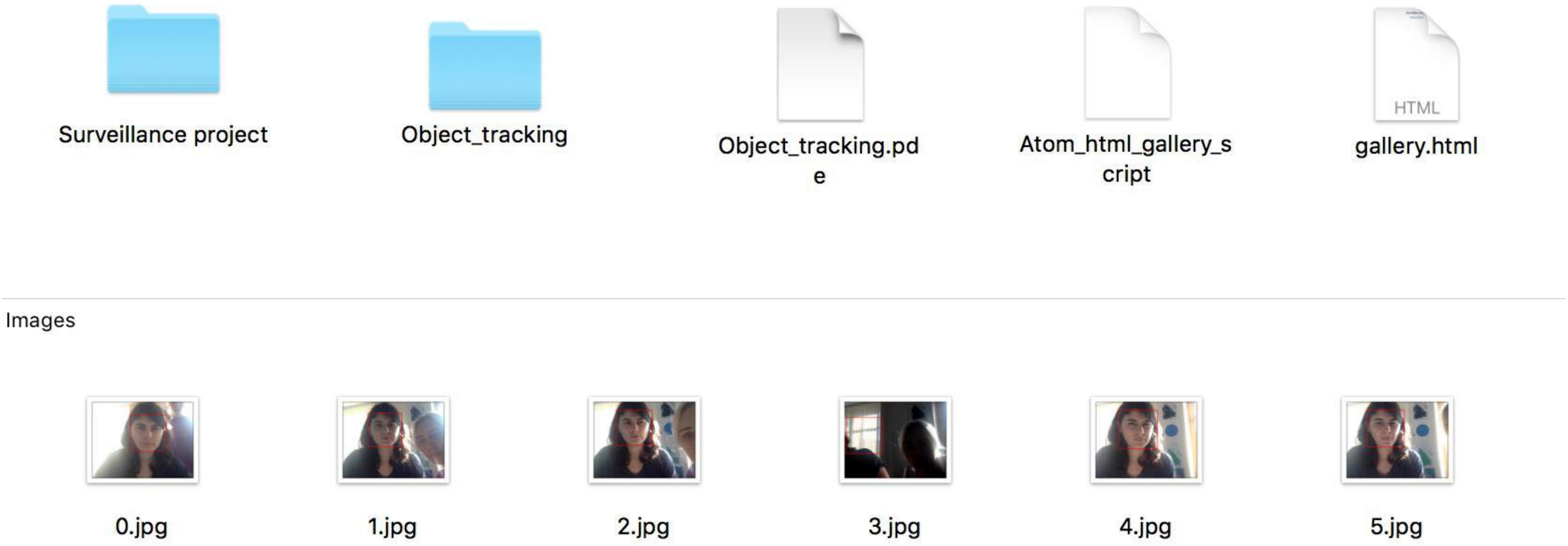
Concept

- This project was created with the aim of exploring the idea of surveillance in a closed space and documenting people's response to it.
- As the surveillance process runs through the laptop's inner camera while the Processing sketch is open, the people present in the room are aware of the fact they are being observed, which invites the viewers to come closer and toy around with the OpenCV effect, making the project somewhat interactive. What the viewers are not aware of is that the inner camera of the computer takes timed screenshots of their faces and sends them over to a website designed in a form of an online image gallery, which serves as a fun data display tool.
- This innovative and entertaining method of displaying data proves amusing for the eye of the viewer and yet raises questions about secret web surveillance and unauthorized public display of private material.
- Question for the class: is it legal and morally correct to display and store pictures of people unaware of being recorded?
- Technical challenge: running the webcam, website and printing simultaneously.

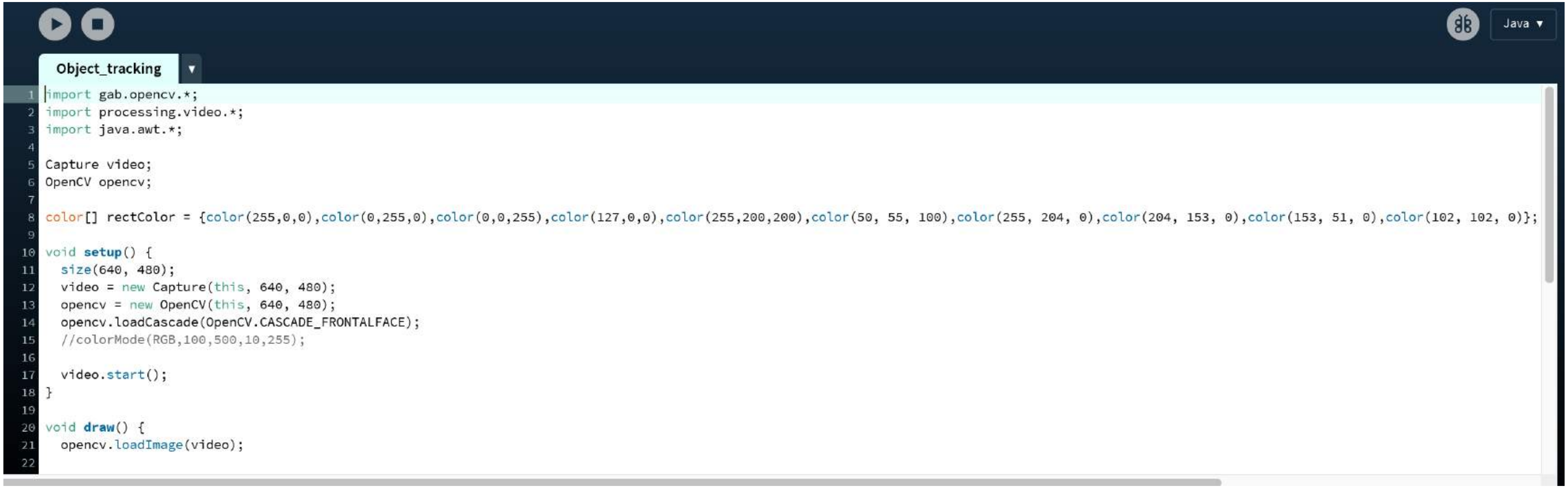
Material

- Laptop MacBook
- Processing libraries (Video / Webcam / OpenCV)
- Atom text editor
- Terminal (Watch script)
- Printer
- A4 paper (Image outcome)

Data Collection



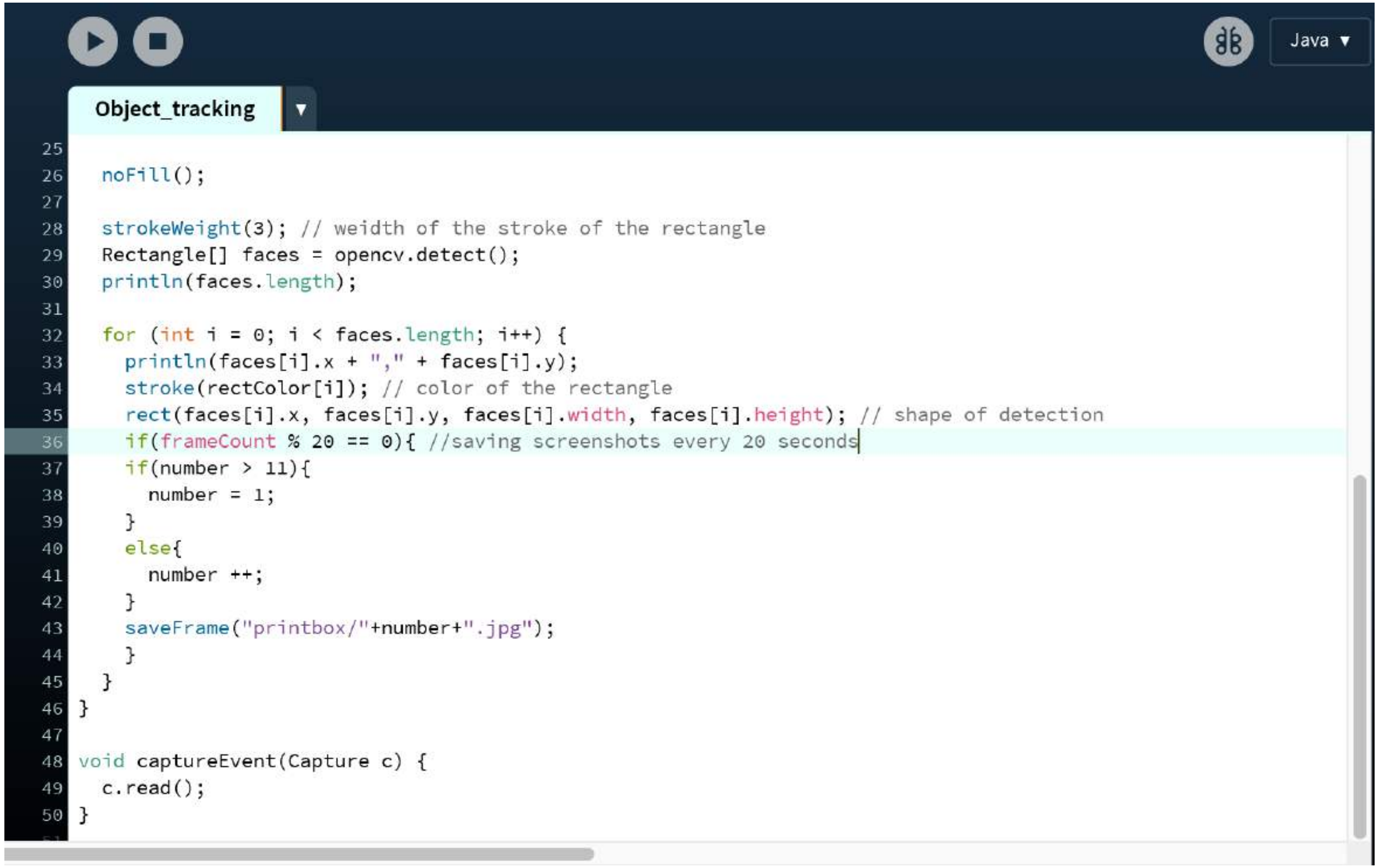
Code: Processing (Part 1)



Object_tracking

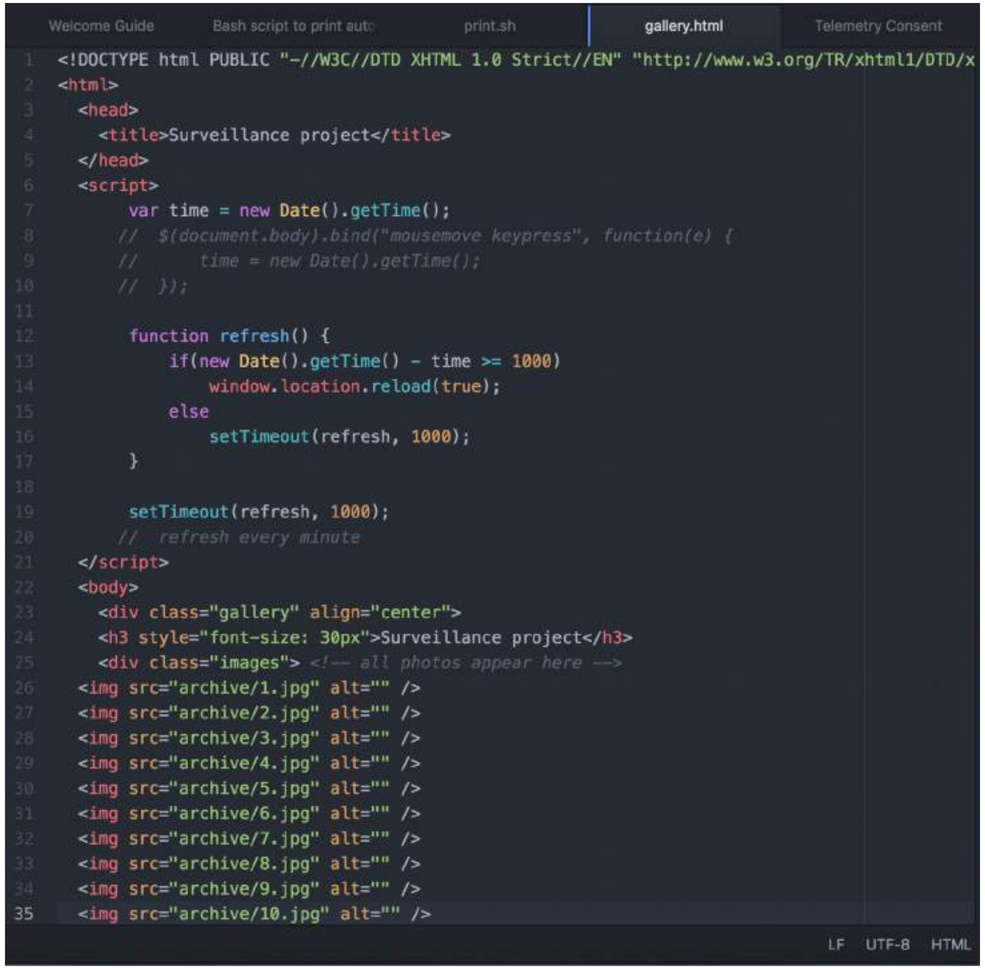
```
import gab.opencv.*;
import processing.video.*;
import java.awt.*;

Capture video;
OpenCV opencv;

color[] rectColor = {color(255,0,0),color(0,255,0),color(0,0,255),color(127,0,0),color(255,200,200),color(50, 55, 100),color(255, 204, 0),color(204, 153, 0),color(153, 51, 0),color(102, 102, 0)};

void setup() {
  size(640, 480);
  video = new Capture(this, 640, 480);
  opencv = new OpenCV(this, 640, 480);
  opencv.loadCascade(OpenCV.CASCADE_FRONTALFACE);
  //colorMode(RGB,100,500,10,255);

  video.start();
}

void draw() {
  opencv.loadImage(video);

```

(Part 2)

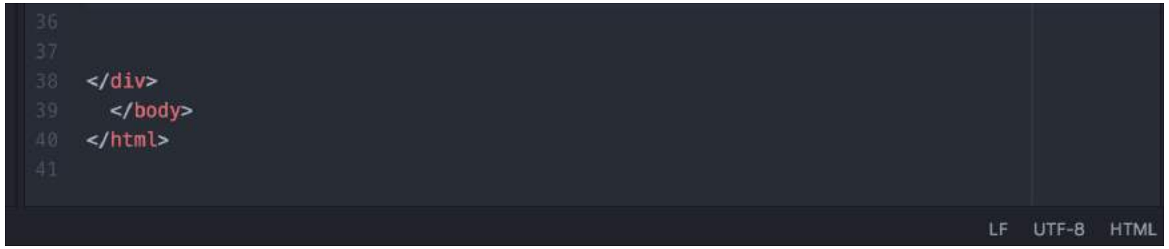


```
  noFill();

  strokeWeight(3); // weidth of the stroke of the rectangle
  Rectangle[] faces = opencv.detect();
  println(faces.length);

  for (int i = 0; i < faces.length; i++) {
    println(faces[i].x + "," + faces[i].y);
    stroke(rectColor[i]); // color of the rectangle
    rect(faces[i].x, faces[i].y, faces[i].width, faces[i].height); // shape of detection
    if(frameCount % 20 == 0){ //saving screenshots every 20 seconds
    if(number > 11){
      number = 1;
    }
    else{
      number ++;
    }
    saveFrame("printbox/"+number+".jpg");
    }
  }
}

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

Code: Atom

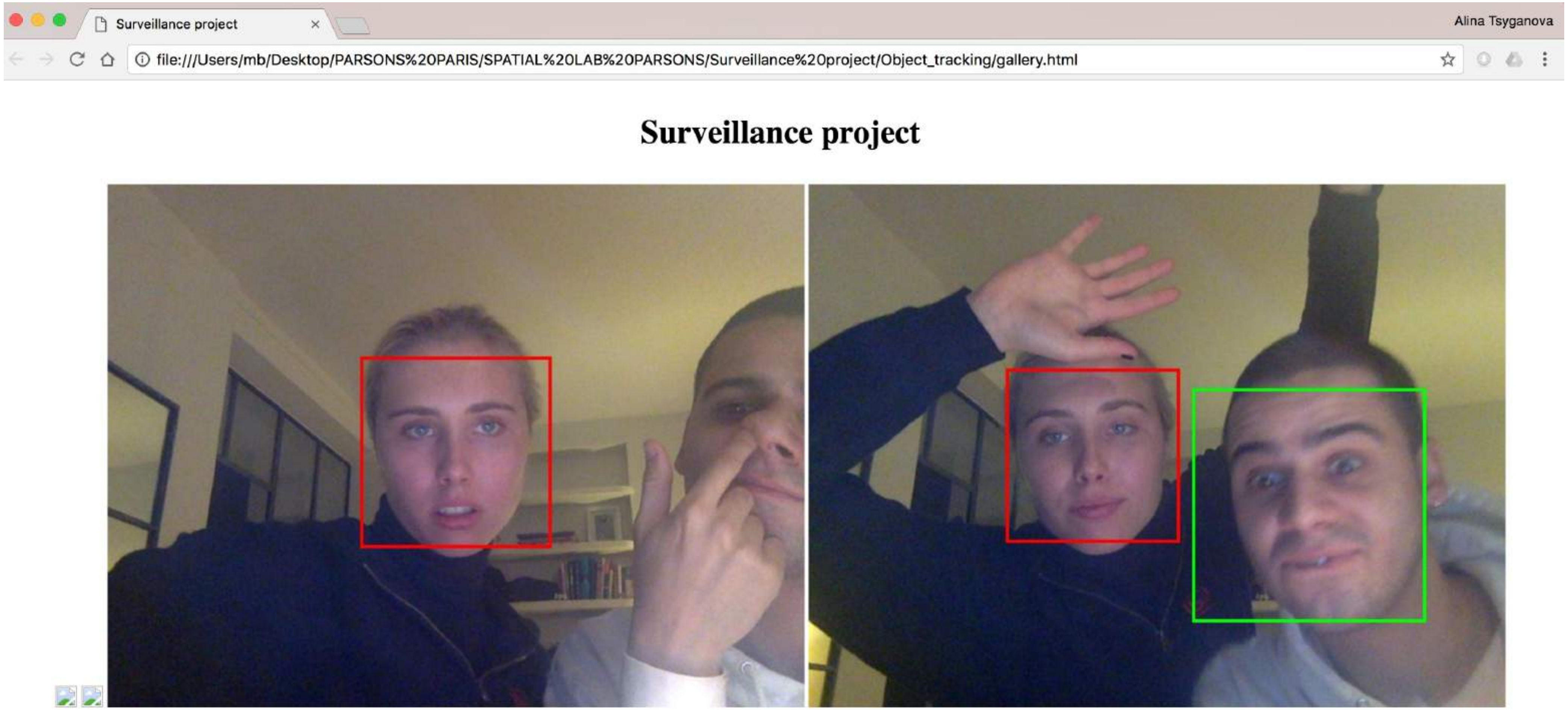


```html
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/x
<html>
  <head>
    <title>Surveillance project</title>
  </head>
  <script>
       var time = new Date().getTime();
      //  $(document.body).bind("mousemove keypress", function(e) {
      //      time = new Date().getTime();
      //  });

       function refresh() {
           if(new Date().getTime() - time >= 1000)
               window.location.reload(true);
           else
               setTimeout(refresh, 1000);
       }

       setTimeout(refresh, 1000);
      //  refresh every minute
  </script>
  <body>
    <div class="gallery" align="center">
    <h3 style="font-size: 30px">Surveillance project</h3>
    <div class="images"> <!-- all photos appear here -->
  <img src="archive/1.jpg" alt="" />
  <img src="archive/2.jpg" alt="" />
  <img src="archive/3.jpg" alt="" />
  <img src="archive/4.jpg" alt="" />
  <img src="archive/5.jpg" alt="" />
  <img src="archive/6.jpg" alt="" />
  <img src="archive/7.jpg" alt="" />
  <img src="archive/8.jpg" alt="" />
  <img src="archive/9.jpg" alt="" />
  <img src="archive/10.jpg" alt="" />
```



```html


</div>
  </body>
</html>

```

Website / HTML Gallery

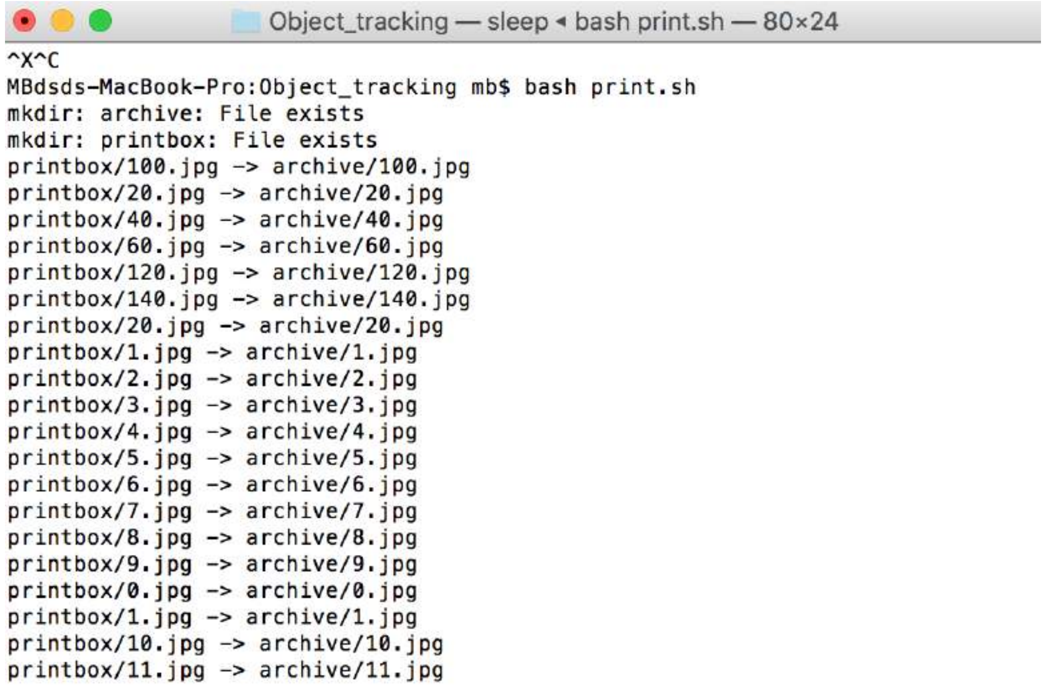


file:///Users/mb/Desktop/PARSONS%20PARIS/SPATIAL%20LAB%20PARSONS/Surveillance%20project/Object_tracking/gallery.html

Surveillance project

Publishing / Printing

- Terminal ‘Watch’ function checks for new pictures in the folder ‘Archive’ in the ‘Object_tracking’ folder every 10 seconds and sends them to print automatically.



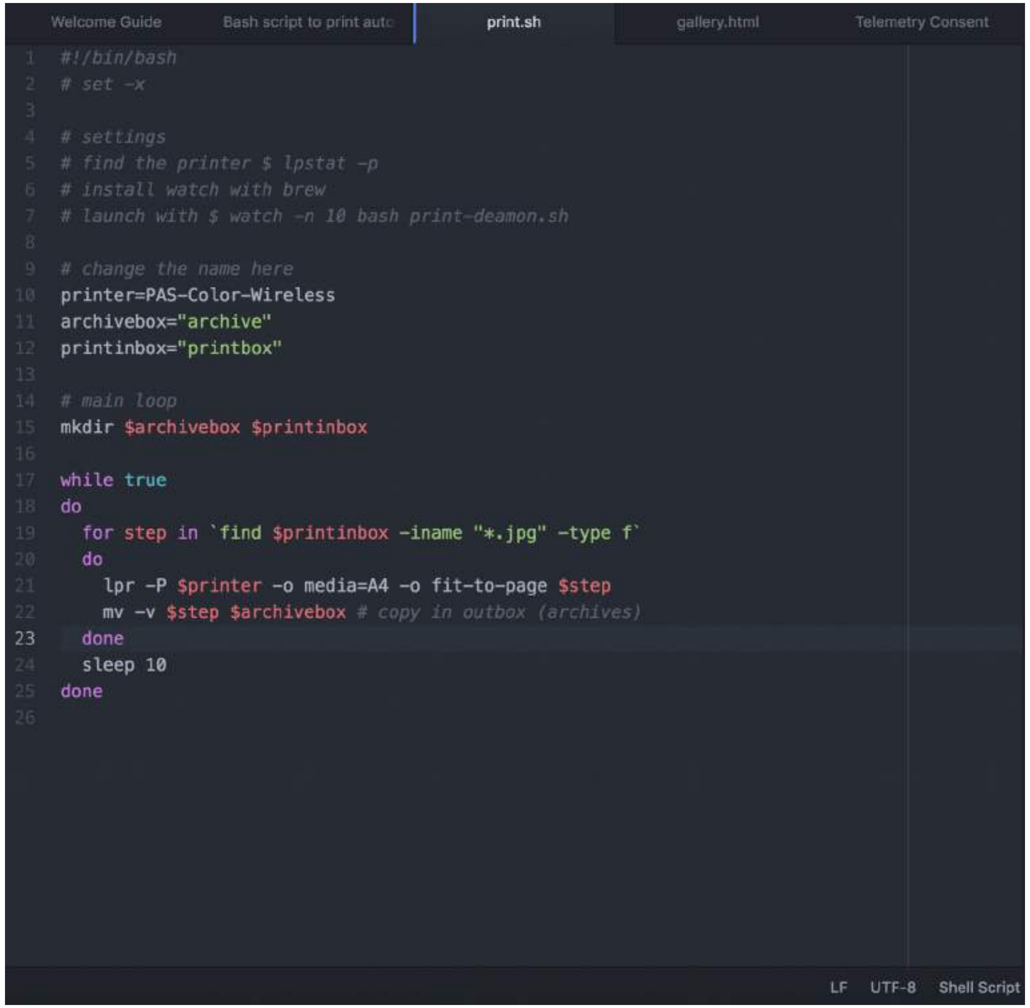
```
^X^C
MBdsds-MacBook-Pro:Object_tracking mb$ bash print.sh
mkdir: archive: File exists
mkdir: printbox: File exists
printbox/100.jpg -> archive/100.jpg
printbox/20.jpg -> archive/20.jpg
printbox/40.jpg -> archive/40.jpg
printbox/60.jpg -> archive/60.jpg
printbox/120.jpg -> archive/120.jpg
printbox/140.jpg -> archive/140.jpg
printbox/20.jpg -> archive/20.jpg
printbox/1.jpg -> archive/1.jpg
printbox/2.jpg -> archive/2.jpg
printbox/3.jpg -> archive/3.jpg
printbox/4.jpg -> archive/4.jpg
printbox/5.jpg -> archive/5.jpg
printbox/6.jpg -> archive/6.jpg
printbox/7.jpg -> archive/7.jpg
printbox/8.jpg -> archive/8.jpg
printbox/9.jpg -> archive/9.jpg
printbox/0.jpg -> archive/0.jpg
printbox/1.jpg -> archive/1.jpg
printbox/10.jpg -> archive/10.jpg
printbox/11.jpg -> archive/11.jpg
```

archive

SHELL

print.sh

Printing code: Atom



```bash
#!/bin/bash
# set -x

# settings
# find the printer $ lpstat -p
# install watch with brew
# launch with $ watch -n 10 bash print-deamon.sh

# change the name here
printer=PAS-Color-Wireless
archivebox="archive"
printinbox="printbox"

# main loop
mkdir $archivebox $printinbox

while true
do
  for step in `find $printinbox -iname "*.jpg" -type f`
  do
    lpr -P $printer -o media=A4 -o fit-to-page $step
    mv -v $step $archivebox # copy in outbox (archives)
  done
  sleep 10
done

```