

```

import processing.serial.*;
import cc.arduino.*;
int maxDays,bright,sat;
float startSecond,degree,degchar,stringlt,ball;
PImage img1;
PImage img2;
PImage img3;
PImage img4;
PImage img5;
PImage img6;
PImage hand;
PImage cure;
PShape vec;
PFont font;

String[] months = new String[12];
Arduino arduino;
//Call the video library
import processing.video.*;
//Capture video from attached camera (webcam)
Capture video;
//Variable for color sampling

int btn=0;
int old_btn=0;
int state=0;
PImage bikini;
PImage tshirt;
PImage hoodie;
PImage coat;
PImage umbrella_1;
PImage umbrella_2;
PImage umbrella_3;

void setup() {
font = loadFont ("Arial-BoldMT-12.vlw");
textFont(font);
smooth();
size(1024,768);
colorMode(HSB,360);
noFill();
strokeWeight(20);
strokeCap(ROUND);
checkMonth();

```

```
startSecond = float(second())*1000.0;
sat=280;
bright=340;
months[0]="January";
months[1]="February";
months[2]="March";
months[3]="April";
months[4]="May";
months[5]="June";
months[6]="July";
months[7]="August";
months[8]="September";
months[9]="October";
months[10]="November";
months[11]="December";
arduino = new Arduino(this, Arduino.list()[0], 57600);
  for (int i = 0; i <= 13; i++)
    arduino.pinMode(i, Arduino.INPUT);
  //Video - What to capture
  video = new Capture(this, width, height, 30);
  cursor(CROSS);
```

```
img1=loadImage("1.png");
img2=loadImage("2.png");
img3=loadImage("3.png");
img4=loadImage("4.jpg");
img5=loadImage("5.jpg");
img6=loadImage("6.jpg");
```

```
hand=loadImage("7.png");
cure=loadImage("8.png");
```

```
textFont(font);
bikini=loadImage("bikini.png");
tshirt=loadImage("tshirt.png");
hoodie=loadImage("hoodie.png");
coat=loadImage("coat.png");
umbrella_1=loadImage("umbrella_1.png");
  umbrella_2=loadImage("umbrella_2.png");
  umbrella_3=loadImage("umbrella_3.png");
  //Anti-aliasing
  smooth();
  //Smooth and fast playback
  frameRate(5);
```

```

}
void draw() {
  int tval=arduino.analogRead(0);
  int lval=arduino.analogRead(1)/4;
  int hval= arduino.analogRead(2)/4;
  int btn =arduino.analogRead(3);
  if((btn==1023)&&(old_btn==0)){
    state=1023-state;}
  old_btn=btn;
  //Capture video
  if (video.available()) {
    video.read();
  }
  //Load video pixels
  video.loadPixels();
  //If the vidOff state is not false, turn the video on
  if(state==1023) {
  display2();
  display();
  }
}

else {image(video,0,0);

background(0);

float worldRecord =500;

//XY coordinate of closest color
int closestX = 0;
int closestY = 0;

//Begin loop to walk through every pixel
for (int x = 0; x < video.width; x ++ ) {
  for (int y = 0; y < video.height; y ++ ) {
    int loc = x + y*video.width;
    //What is current color
    color currentColor = video.pixels[loc];
    float r1 = red(currentColor);
    float g1 = green(currentColor);
    float b1 = blue(currentColor);
    // Using euclidean distance to compare colors
    float d = dist(r1,g1,b1,214,80,121);
    // We are using the dist( ) function to compare the current color with the color we are tracking.
    // If current color is more similar to tracked color than

```

```

// closest color, save current location and current difference
if (d < worldRecord) {
    worldRecord = d;
    closestX = x;
    closestY = y;
}
}
}

if(worldRecord<10){

if (tval>=0&&tval<=64){image(coat,closestX-400,closestY-100,800,640);}
else if(tval>64&&tval<=128){image(hoodie,closestX-400,closestY-100,800,640);}
else if(tval>128&&tval<=192){image(tshirt,closestX-400,closestY-100,800,640);}
else{image(bikini,closestX-400,closestY-100,800,640);}
    if (hval>=0&&hval<=90){image(umbrella_3,closestX-800,closestY-100,800,640);}
else if (hval>90&&hval<=130){image(umbrella_2,closestX-800,closestY-100,800,640);}
else {image(umbrella_1,closestX-800,closestY-100,800,640);}

    glasses(closestX-20,closestY+20,abs(lval-255));    }

}
println(hval);}

void glasses(int x,int y,int z){

    ellipseMode(CENTER);
    rectMode(CENTER);
    fill (0,99,99);

    ellipse (x,y, 80, 80);
    ellipse(x+70, y, 80, 80);

    rect(x+20,y-12,60,40);
    fill (z);
    ellipse (x,y, 60,60);
    ellipse(x+70, y, 60, 60);
}

void display(){
    pushMatrix();
    translate(width/2+300,height/2+200);
    rotate(radians(-90));
}

```

```

degree = lerp(0,360,norm((startSecond+millis())%60000,0,60000));
stroke(degree,sat,bright);
arc(0, 0, 285, 285, radians(4), radians(degree));

writeOnArc(second()+" seconds",-139, degree);

degree = lerp(0,360,norm((minute()*60+second())%3600,0,3600));
stroke(degree,sat,bright);
arc(0, 0, 240, 240, radians(5), radians(degree) );

writeOnArc(minute()+" minutes",-117, degree);

degree = lerp(0,360,norm((hour()*60+minute())%1440,0,1440));
stroke(degree,sat,bright);
arc(0, 0, 195, 195, radians(6), radians(degree) );

writeOnArc(hour()+" hours",-95, degree);

degree = lerp(0,360,norm(day(),0,maxDays));
stroke(degree,sat,bright);
arc(0, 0, 130, 130, radians(9), radians(degree));

writeOnArc(day()+" day",-62, degree);

degree = lerp(0,360,norm(month(),0,12));
stroke(degree,sat,bright);
arc(0, 0, 85, 85, radians(13), radians(degree));

writeOnArc(months[month()-1],-39, degree);

noStroke();

ball = (sin(radians(millis()*0.06)))*260+100;
if(ball>35)
  fill(0,10,ball);
else
  fill(0,10,35);

ellipse(0,0,30,30);
noFill();
popMatrix();}

void checkMonth(){
  if(month()==1 || day()==3 || day()==5 || day()==7 || day()==8 || day()==10 || day()==12)

```

```

    maxDays=31;
    if(month()==2)
        maxDays=28;
    else
        maxDays=30;
}

void writeOnArc(String a, float ypos, float rot){
    pushMatrix();
    rotate(radians(93));
    stringlt = textWidth(a);
    degchar = (stringlt * 360) / (ypos * TWO_PI);
    rotate(radians(degchar));
    rotate(radians(rot));
    for(int i=0;i<a.length();i++)
    {
        fill(0,10,35);
        text(a.charAt(i),0,ypos);
        rotate(radians((textWidth(a.charAt(i))*-360)/(ypos*TWO_PI)));
        noFill();
    }
    popMatrix();
}

void display2(){

    pushMatrix();
    float rrr=random(-180, 360);
    float rrrr=random(255);
    float rrrr1=random(255);
    float rrrr2=random(255);
    float rrrr3=random(50, 100);
    float sca=random(.2, 2);

    rotate(rrr);
    //tint(rrrr, rrrr1, rrrr2, rrrr3);
    scale(sca);
    image(hand, 500, 400, 500, 500);

    popMatrix();

    pushMatrix();
    float xrrr=random(-180, 360);

```

```
float xrrrr=random(255);
float xrrrr1=random(255);
float xrrrr2=random(255);
float xrrrr3=random(50, 100);
float xsca=random(.2, 2);
```

```
rotate(xrrrr);
//tint(xrrrr, xrrrr1, xrrrr2, xrrrr3);
scale(xsca);
image(cure, 500, 200, 500, 500);
  popMatrix();
float r =random(10, 13);
```

```
float x1= random(3*PI);
float x2= random(3*PI);
float x3= random(3*PI);
float x4= random(3*PI);
float x5= random(3*PI);
```

```
float y1= random(1, 2);
float y2= random(1, 2);
float y3= random(1, 2);
float y4= random(1, 2);
float y5= random(1, 2);
```

```
float c1= random(200);
float c2= random(200);
float c3= random(200);
float c4= random(200);
float c5= random(200);
```

```
float cc1= random(200);
float cc2= random(200);
float cc3= random(200);
float cc4= random(200);
float cc5= random(200);
```

```
float t= random(255);
float t1= random(255);
```

```
float t2= random(255);  
float t3= random(50, 130);
```

```
//tint(t, t1, t2, t3);
```

```
pushMatrix();
```

```
float q= random(1, 100);  
float a= random(-60, 0);  
float a2= random(-60, 0);
```

```
scale(q);  
image(img6, a, a2);  
//filter(BLUR);  
popMatrix();
```

```
pushMatrix();  
rotate(x1);  
scale(y1);  
translate(c1, cc1);  
image(img1, r, r);  
popMatrix();
```

```
pushMatrix();  
rotate(x2);  
scale(y1);  
translate(c2, cc2);  
image(img2, r, r);  
popMatrix();
```

```
pushMatrix();  
rotate(x3);  
scale(y1);  
translate(c3, cc3);  
image(img3, r, r);  
popMatrix();
```

```
pushMatrix();  
rotate(x4);  
scale(y1);  
translate(c4, cc4);
```



```
image(img4, r, r);  
popMatrix();
```

```
pushMatrix();  
rotate(x5);  
scale(y1);  
translate(c5, cc5);  
image(img5, r, r);  
popMatrix();  
}
```