Eye-Tracking System

NEW MEDIA STUDIES - PROJECT

BY: AISHANI GOEL- 2K19/BD/04

The Code

1ST VERSION

```
let capture = null;
let tracker = null;
let positions = null;
let w = 0, h = 0;

function setup() {
  w = windowWidth;
  h = windowHeight;
  capture = createCapture(VIDEO);
  createCanvas(w, h);
  capture.size(w, h);
  capture.hide();
```

//DECLARING VARIABLES AND THEIR BASIC VALUES

//TO CAPTURE VIDEO FROM THE WEBCAM AND DISPLAY ON THE CANVAS AS WELL WITH INVERT FILTER. BY DEFAULT, THE CAPTURE FEED SHOWS UP WHICH IS HIDDEN BY CAPTURE.HIDE()

```
tracker = new clm.tracker();
 tracker.init();
 tracker.start(capture.elt);
function draw() {

 translate(w, 0);
 scale(-1.0, 1.0);
  positions = tracker.getCurrentPosition();

 if (positions.length > 0) {

   const eye1 = {
     outline: [23, 63, 24, 64, 25, 65, 26, 66].map(getPoint),
     center: getPoint(27),
     top: getPoint(24),
     bottom: getPoint(26)
   };
```

//CLM.TRACKER IS JS LIBRARY FOR FITTING FACIAL MODELS TO FACES IN IMAGES AND VIDEO

//INIT(MODEL) : INITIALIZE CLMTRACKR.

// FLIP THE CANVAS SO THAT WE GET A MIRROR IMAGE

//GET THE CURRENT POSITIONS OF THE FITTED FACIAL MODEL

// EYE POINTS FROM CLMTRACKR: //HTTPS://WWW.AUDUNO.COM/CLMTRACKR/DOCS/REFERENCE.HTML

```
  const eye2 = {
    outline: [28, 67, 29, 68, 30, 69, 31, 70].map(getPoint),
    center: getPoint(32),
    top: getPoint(29),
    bottom: getPoint(31)
  }

  const irisColor = color(random(360), 80, 80, 0.4);
  drawEye(eye1, irisColor);
 drawEye(eye2, irisColor);
 }
}

function getPoint(index) {
 return createVector(positions[index][0], positions[index][1]);
}
```

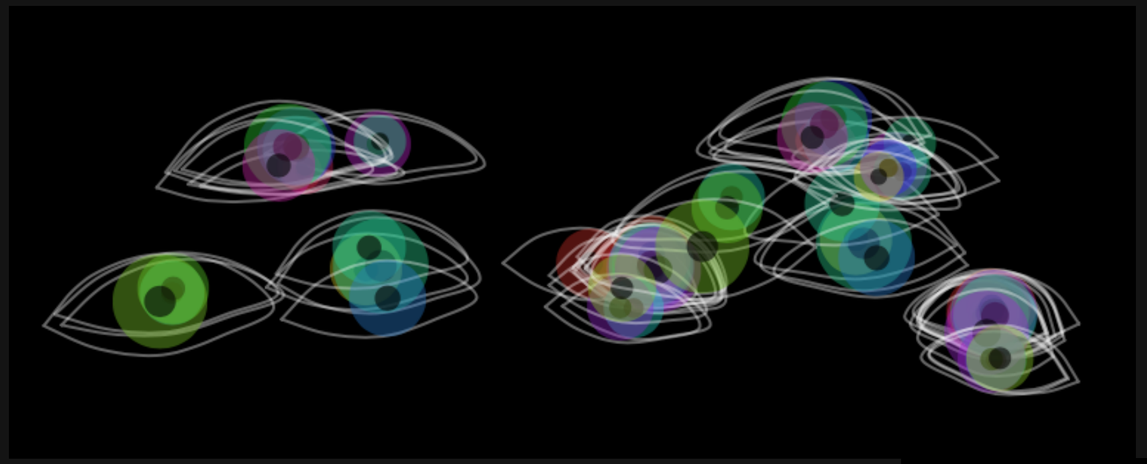
// EYE POINTS FROM CLMTRACKR:
//HTTPS://WWW.AUDUNO.COM/CLMTRACKR/DOCS/REFERENCE.HTML

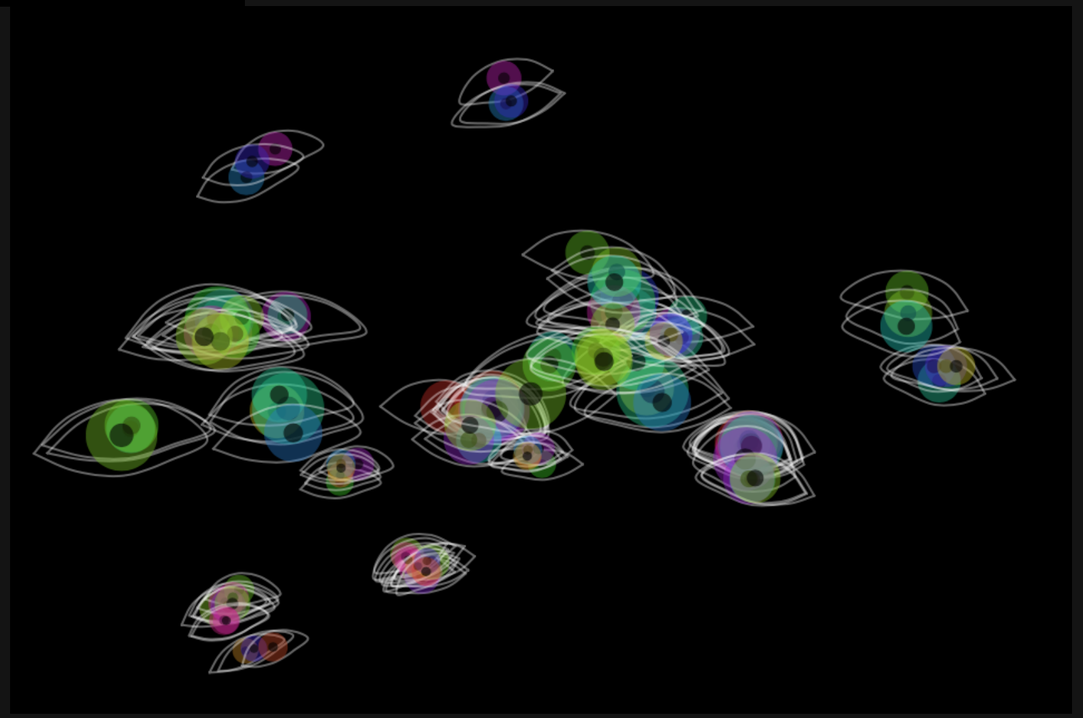
//DRAW THE CURRENTLY FITTED FACIAL MODEL

```
function drawEye(eye, irisColor) {
  noFill();
  stroke(255, 0.4);
  drawEyeOutline(eye);                    //DRAW THE OUTILINE OF EYE

  const irisRadius = min(eye.center.dist(eye.top), eye.center.dist(eye.bottom));
  const irisSize = irisRadius * 2;
  noStroke();
  fill(irisColor);
  ellipse(eye.center.x, eye.center.y, irisSize, irisSize);

  const pupilSize = irisSize / 3;                    //THE SIZE OF THE EYE AND IRIS CHANGES
  fill(0, 0.6);
  ellipse(eye.center.x, eye.center.y, pupilSize, pupilSize);
}
```

```
function drawEyeOutline(eye) {
beginShape();
  const firstPoint = eye.outline[0];
  eye.outline.forEach((p, i) => {
    curveVertex(p.x, p.y);
    if (i === 0) {
      curveVertex(firstPoint.x, firstPoint.y);      //DUPLICATE THE INITIAL POINT
    }
    if (i === eye.outline.length - 1) {
          curveVertex(firstPoint.x, firstPoint.y);
      curveVertex(firstPoint.x, firstPoint.y);      // CLOSE THE CURVE AND
    }                                               DUPLICATE THE CLOSING POINT
  });
  endShape();
}
```

```
function keyPressed() {
  // Clear background       //CLEARING OF CANVAS BY A MOUSE CLICK
  background(0);
}
function mouseClicked() {
  const timestamp = timestampString();                    //TO SAVE THE IMAGE
  saveCanvas("eyeTrail-" + timestamp, "png");
}
function timestampString() {
  return year() + nf(month(), 2) + nf(day(), 2) + "-" + nf(hour(), 2) +
nf(minute(), 2) + nf(second(), 2);
}
function windowResized() {
  w = windowWidth;
  h = windowHeight;
  resizeCanvas(w, h);
  background(0);
}
```





SCREENSHOTS OF THE OUTPUT

The Code

2ND VERSION

*THIS PROGRAM IS INCOMPLETE AS IT SHOWS RUN-TIME ERROR.

```
var webcam = null;
var tracker = null;
var features = null;          //LIST OF FACIAL FEATURES
function setup() {
  createCanvas(640, 480);

  webcam = createCapture(VIDEO);
  webcam.size(width, height);                  //TO CREATE WEBCAM
  webcam.hide();
```

The Code

2ND VERSION

```
tracker = new clm.tracker();
  tracker.init();                          //CONNECT FACE TRACKING TO WEBCAM
  tracker.start(webcam.elt);
}

function draw() {

  translate(width,0);
  scale(-1,1);
  image(webcam, 0,0);
  features = tracker.getCurrentPosition();   //GET THE CURRENT POSITIONS OF THE
                                             FITTED FACIAL MODEL
```

The Code

2ND VERSION

```
if(features.length>0) {

  var leftEye = features[27];
  var leftEyeX = features[0];                    //DEFINING OF EYE VARIABLES
  var leftEyeY = features[1];
     fill(0);
  stroke(255);
  ellipse(leftEyeX, leftEyeY, 10, 10);
     var rightEyeX = features[32][0];
  var rightEyeY = features[32][1];
  ellipse(rightEyeX, rightEyeY, 10, 10);
  }
}
```

Usage

ARTWORK

Creating artwork using your eyes

RESEARCH PROJECTS

To track your eye movements for research purposes

ANALYSIS

Analysing the data collected through this program