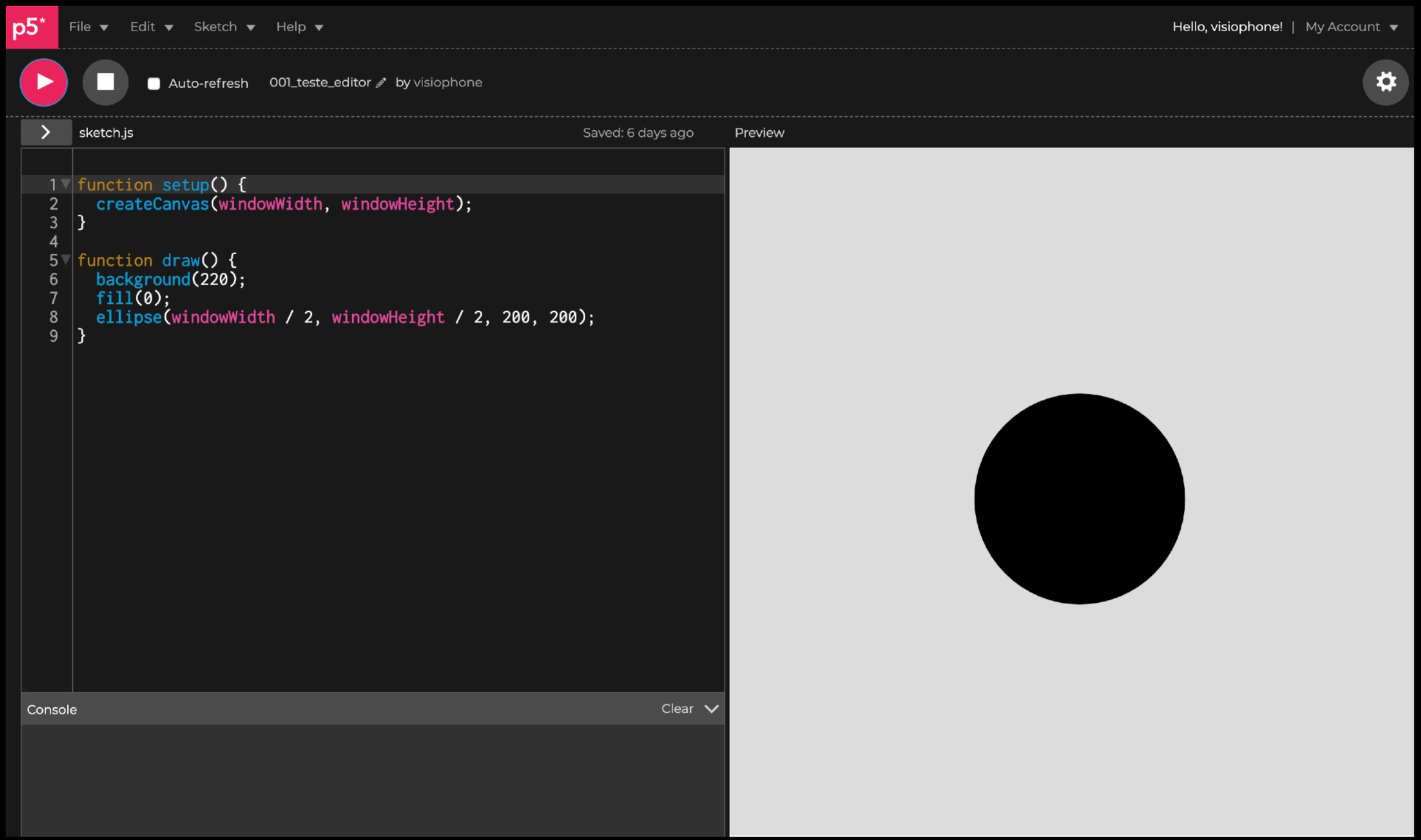
P5.js Body as Interface

Rodrigo Carvalho (@VISIOPHONE_)
Processing Community Day Porto 2024

CODE COLLECTION AT P5JS EDITOR

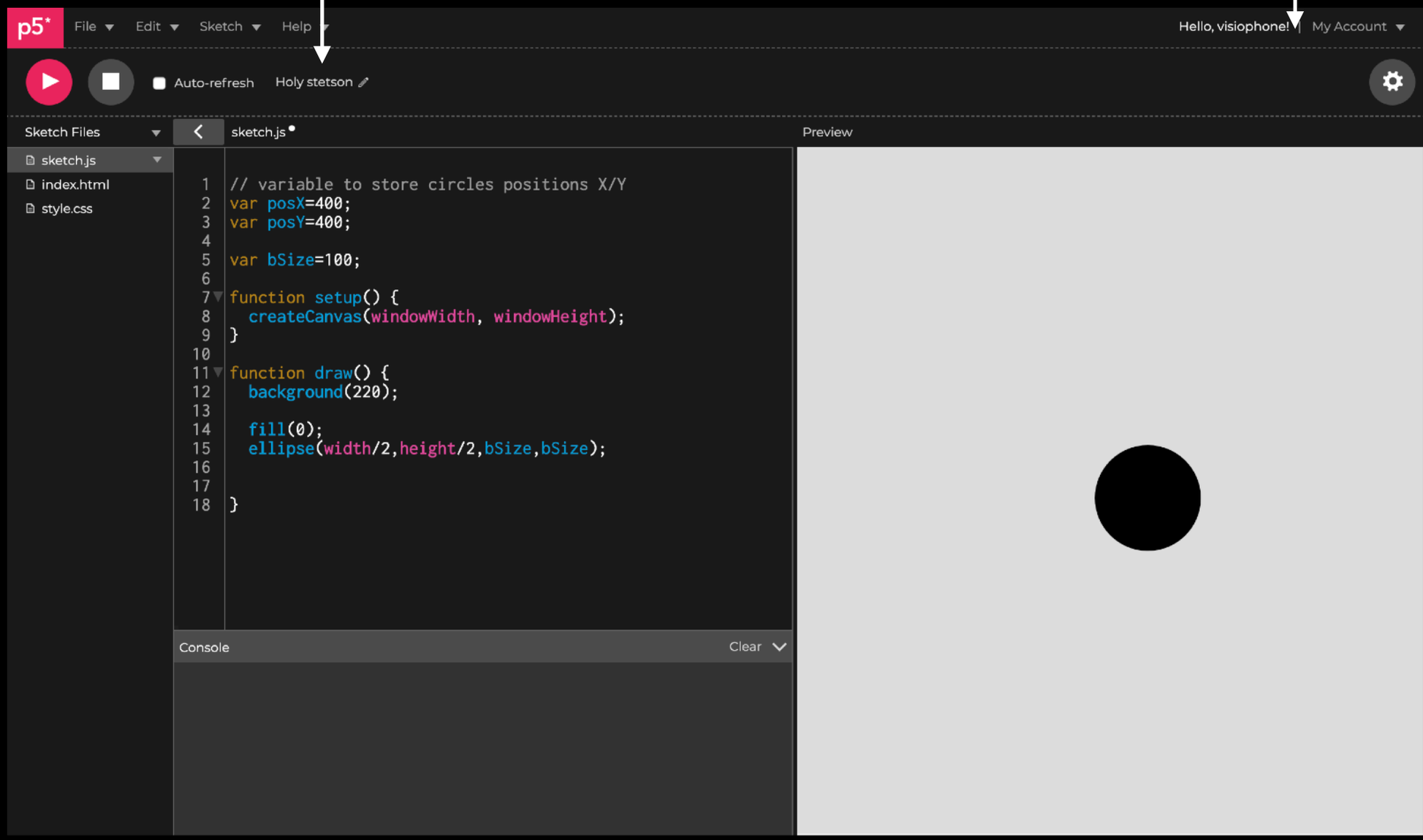
P5JS (Processing / javascript / web / browser)

P5JS web.editor: editor.p5js.org



Account
Collections
Sketchs

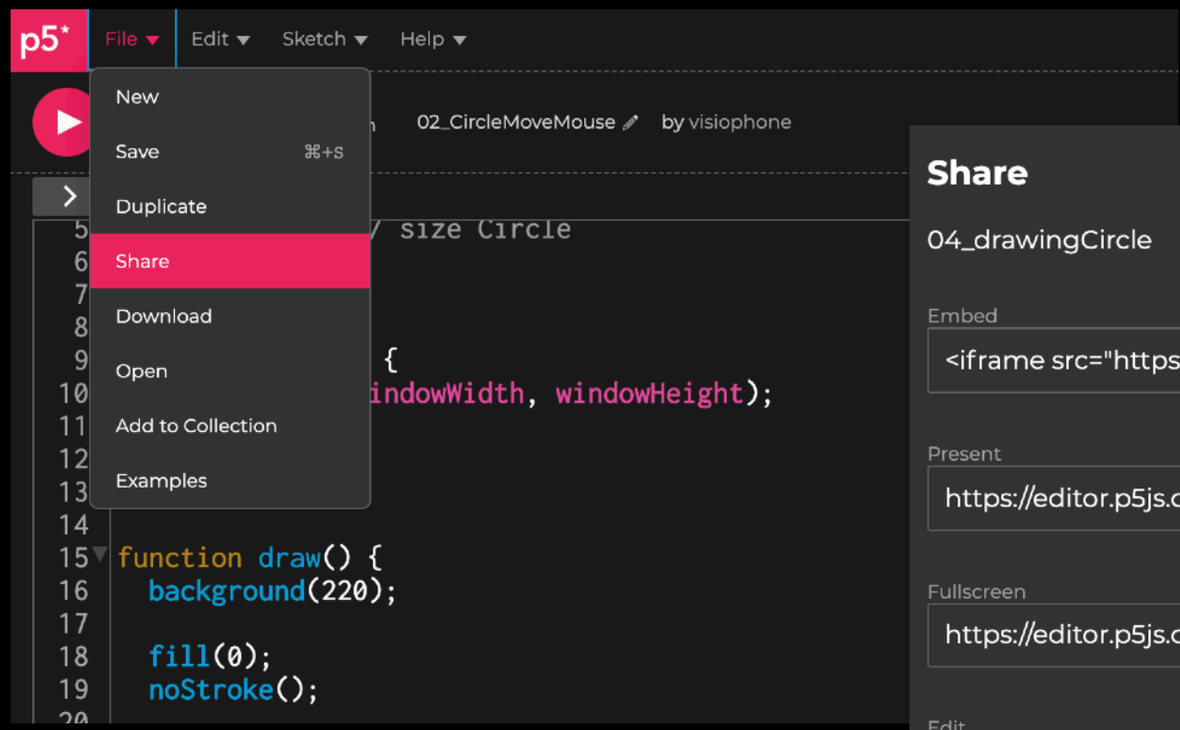
FILE NAME

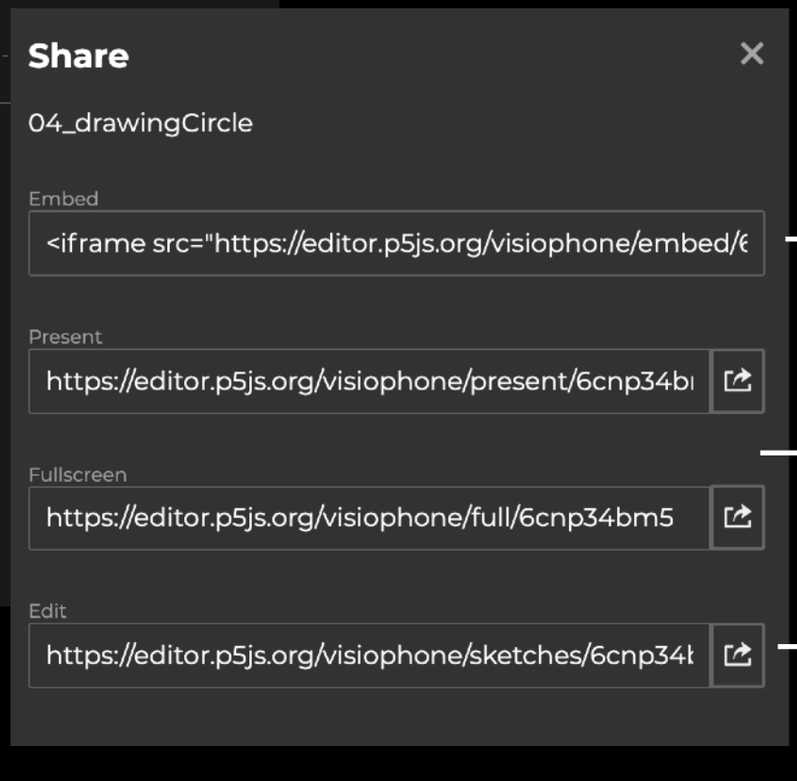


FILES
WEB

CONSOLE / PRINT

PREVIEW

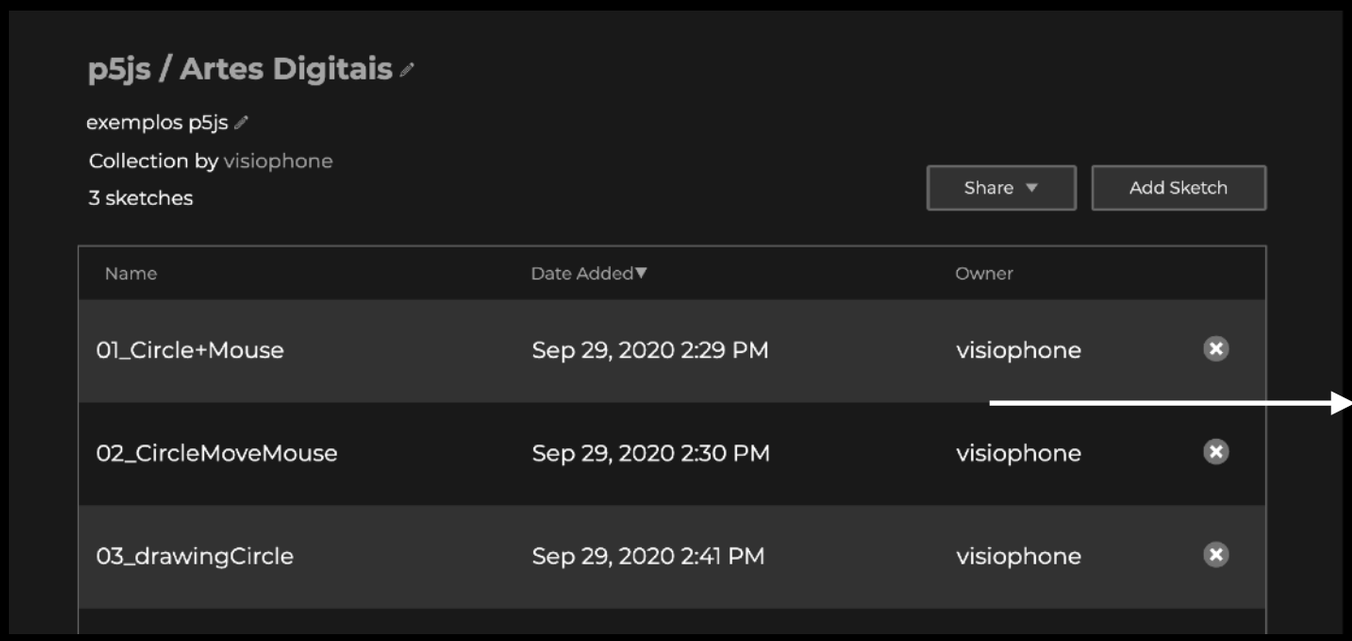




Embed website

Full screen

Share with editable code



COLLECTIONS

DIFFERENCES P5JS - PROCESSING
HTTPS://GITHUB.COM/PROCESSING/P5.JS/WIKI/PROCESSING-TRANSITION

```
1  // variable to store circles positions X/Y
2  var posX=400;
3  var posY=400;
4
5  var bSize=100;
6
7  function setup() {
8    createCanvas(windowWidth, windowHeight);
9  }
10
11 function draw() {
12   background(220);
13
14   fill(0);
15   ellipse(width/2,height/2,bSize,bSize);
16
17
18 }
```

```
float posY=400;

float bSize=100;

void setup() {
  size(displayWidth, displayHeight);
}

void draw() {
  background(220);

  fill(0);
  ellipse(width/2,height/2,bSize,bSize);


}
```

p5.js

a cheat sheet for beginners!

program structure

```
//runs once when program starts
function setup(){
  createCanvas(800,600);
}

//run continuously after setup
function draw(){
  //rendering loop
}
```

system variables

windowWidth / windowHeight
width / height of window

width / height
width / height of canvas

mouseX / mouseY
current horizontal / vertical mouse position

non-visual feedback

print()
report data to the output console

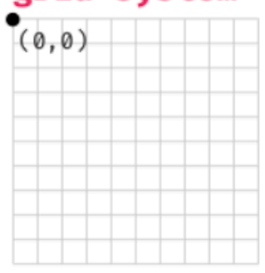
color

fill(120) gray: 0-255
fill(100,125,255) r, g, b: 0-255
fill(255, 0, 0, 50) r, g, b, alpha
fill('red') color string
fill('#ccc') 3-digit hex
fill('#222222') 6-digit hex fill
color(0, 0, 255) p5.Color object

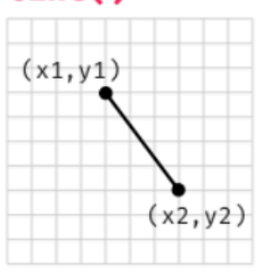
2d primitives

```
line(x1, y1, x2, y2)

ellipse(x1, y1, width, height)

rect(x1, y1, width, height)

arc(x1, y1, width, height, start, stop)

beginShape();
  vertex(x1, y1);
  vertex(x2, y2);
  vertex(x3, y3);
  //add more vertex
endShape(CLOSE);

text("string", x, y, boxwidth, boxheight)
```

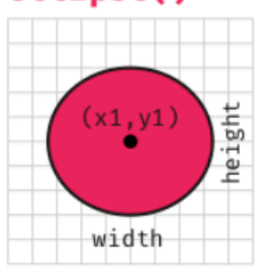
grid system



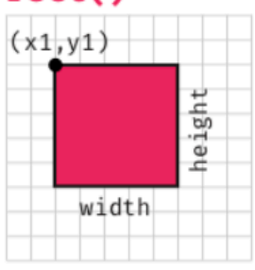
line()



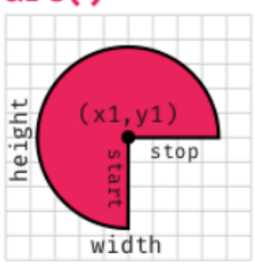
ellipse()



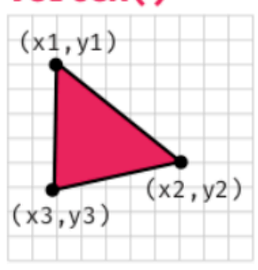
rect()



arc()



vertex()



math

```
+ - / *

random(low,high)

map(value, in1, in2, out1, out2)
map a value from input to output range
```

attributes

background(color)
set the background color

fill(color)
set the fill color

noFill()
disables fill

stroke(color)
set the stroke color

strokeWeight(weight)
set the stroke's width

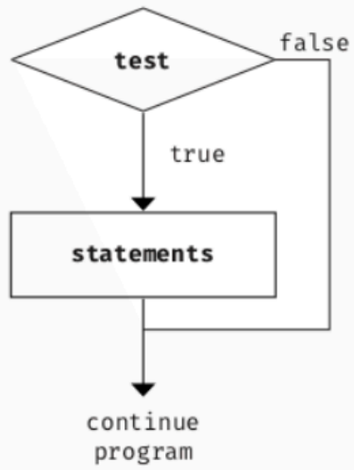
noStroke()
disables stroke

ellipseMode(MODE)
rectMode(MODE)
CENTER,CORNER

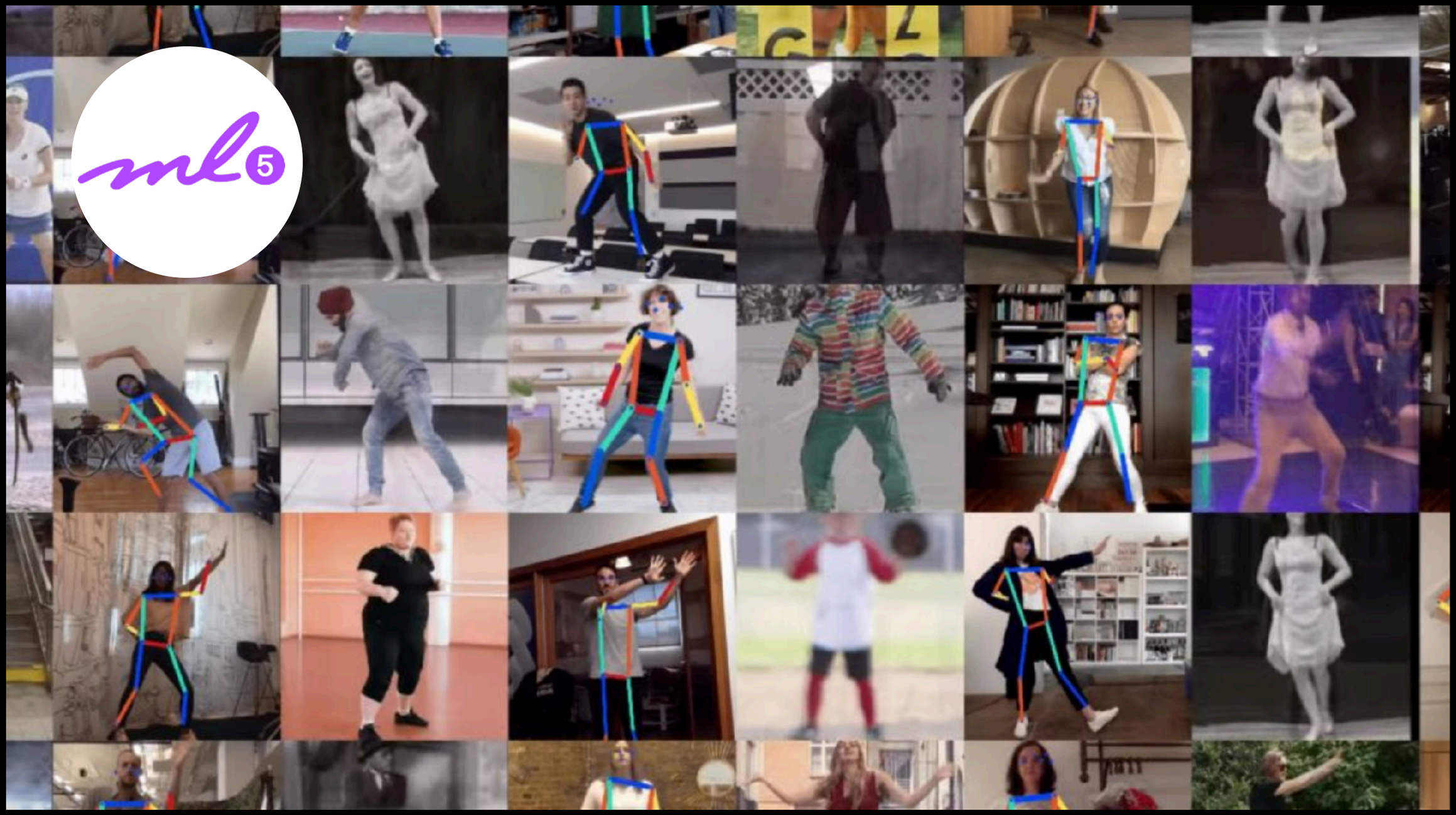
textSize(pixels)

if/then logic

```
if(test){
  statements
}
```



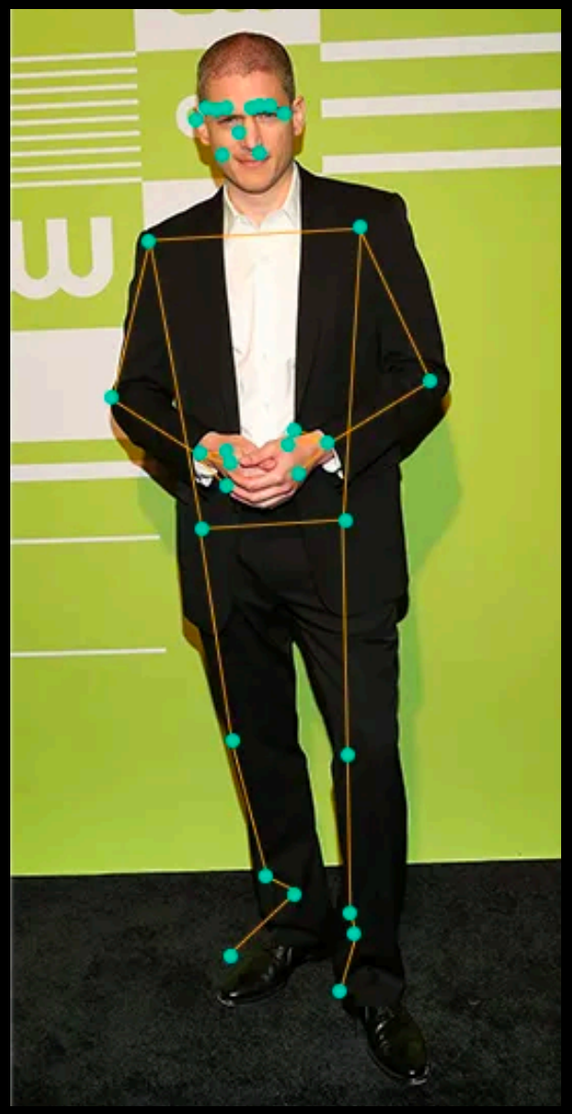

```
== equal to
!= not equal
>  greater than
<  less than
>= greater than or equal
<= less than or equal
```

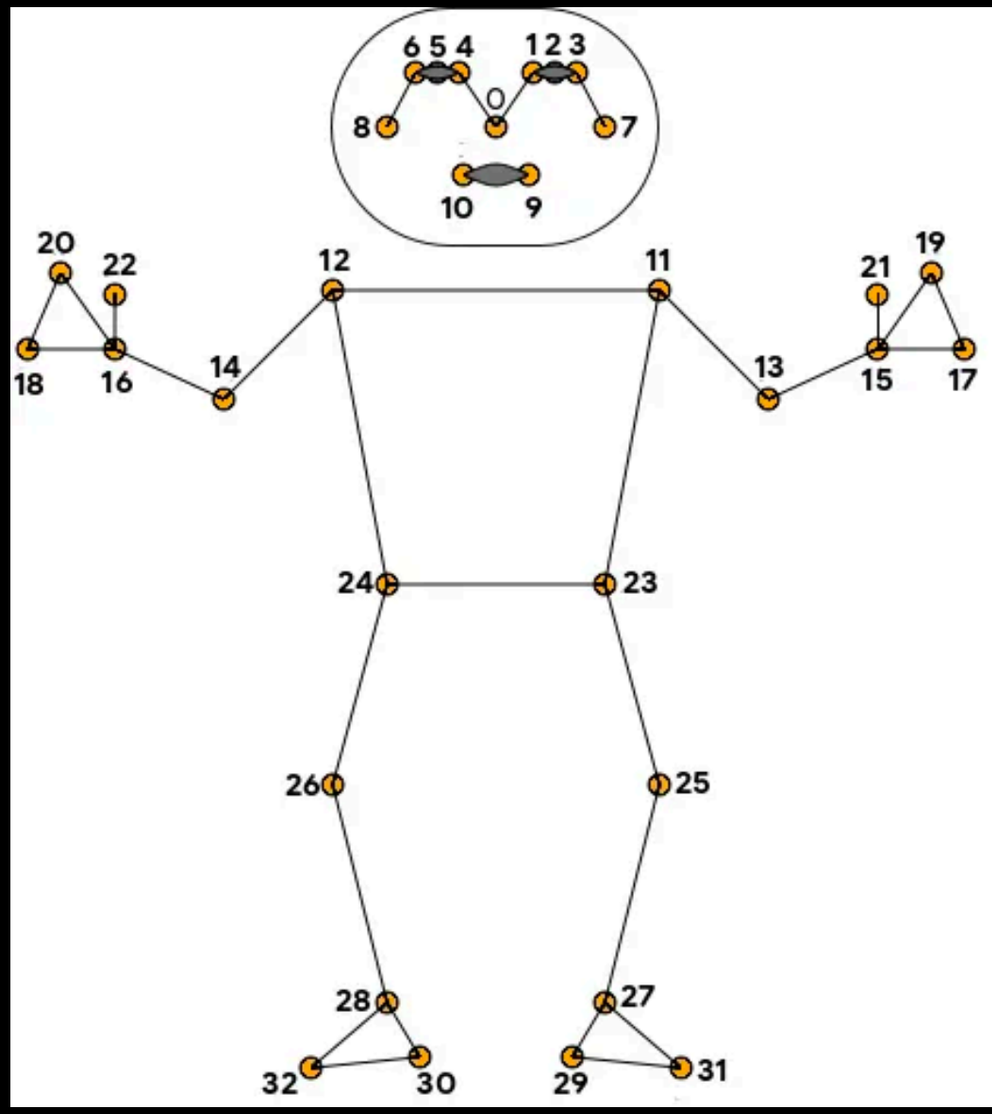


ML5 | a library/platform that provides access to machine learning algorithms and models in the browser

https://ml5js.org/ Examples ML5 p5js-Editor

BLAZEPOSE : Machine-learning model for Real-time Human Pose Estimation.

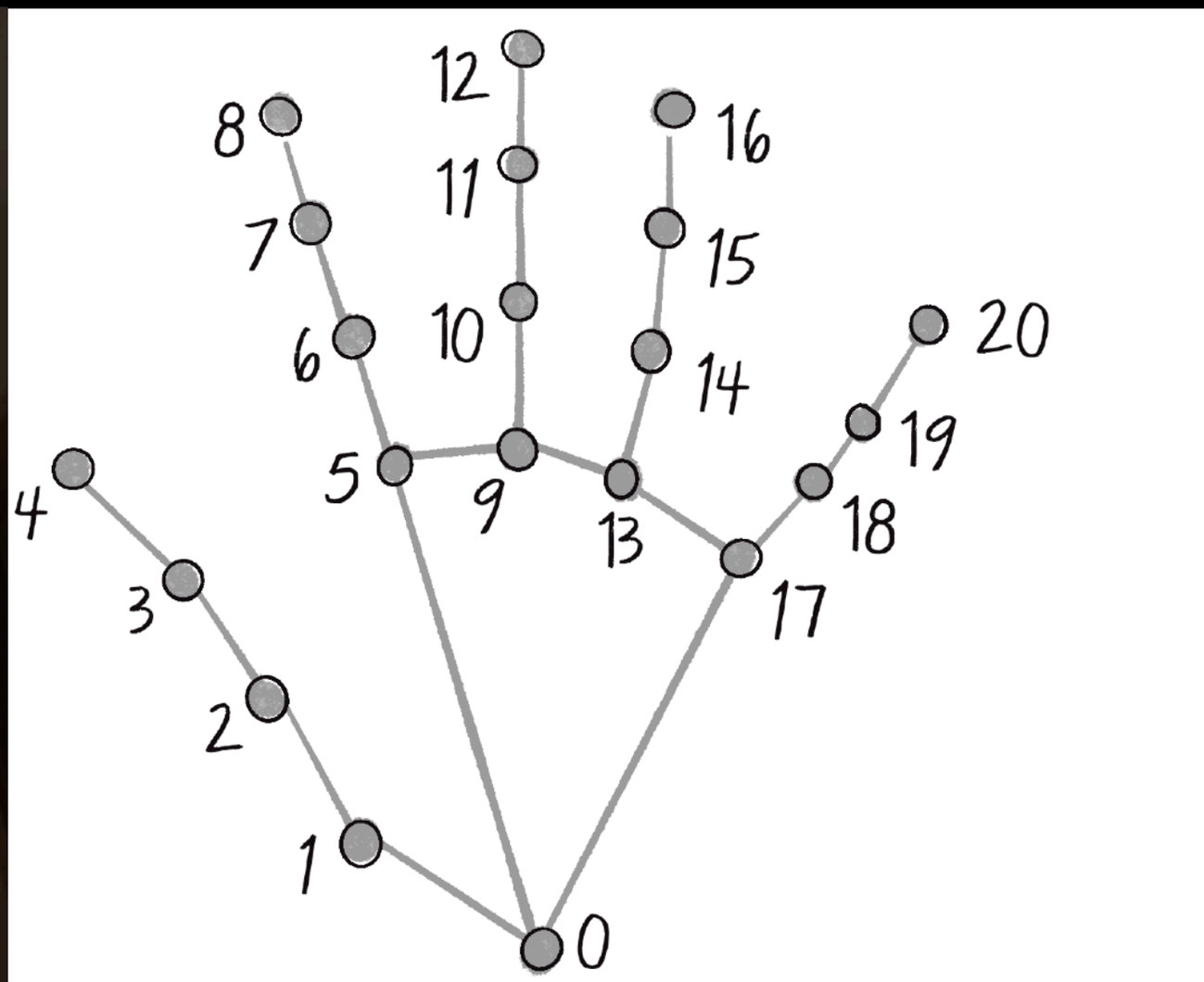




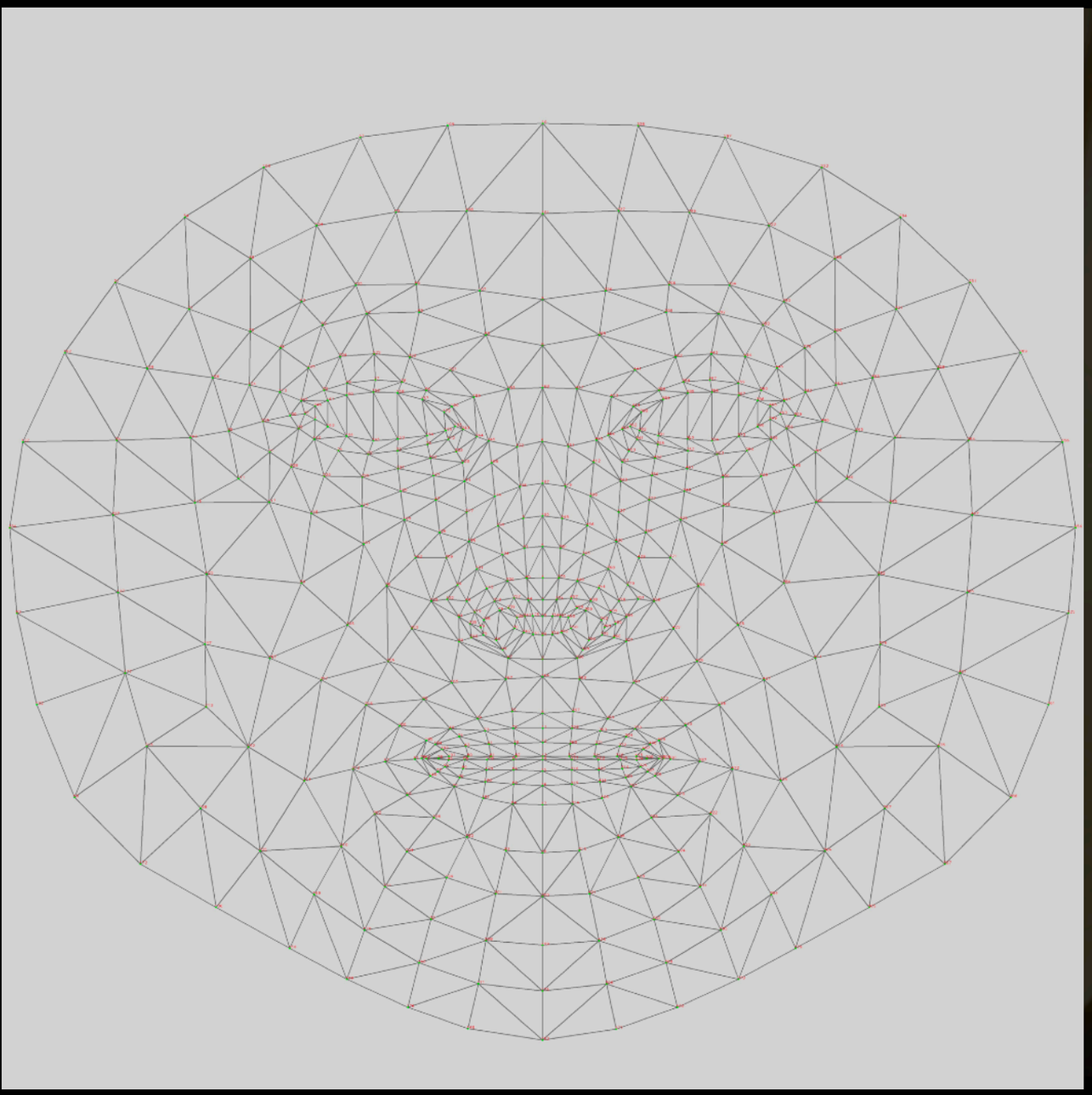
Landmarks

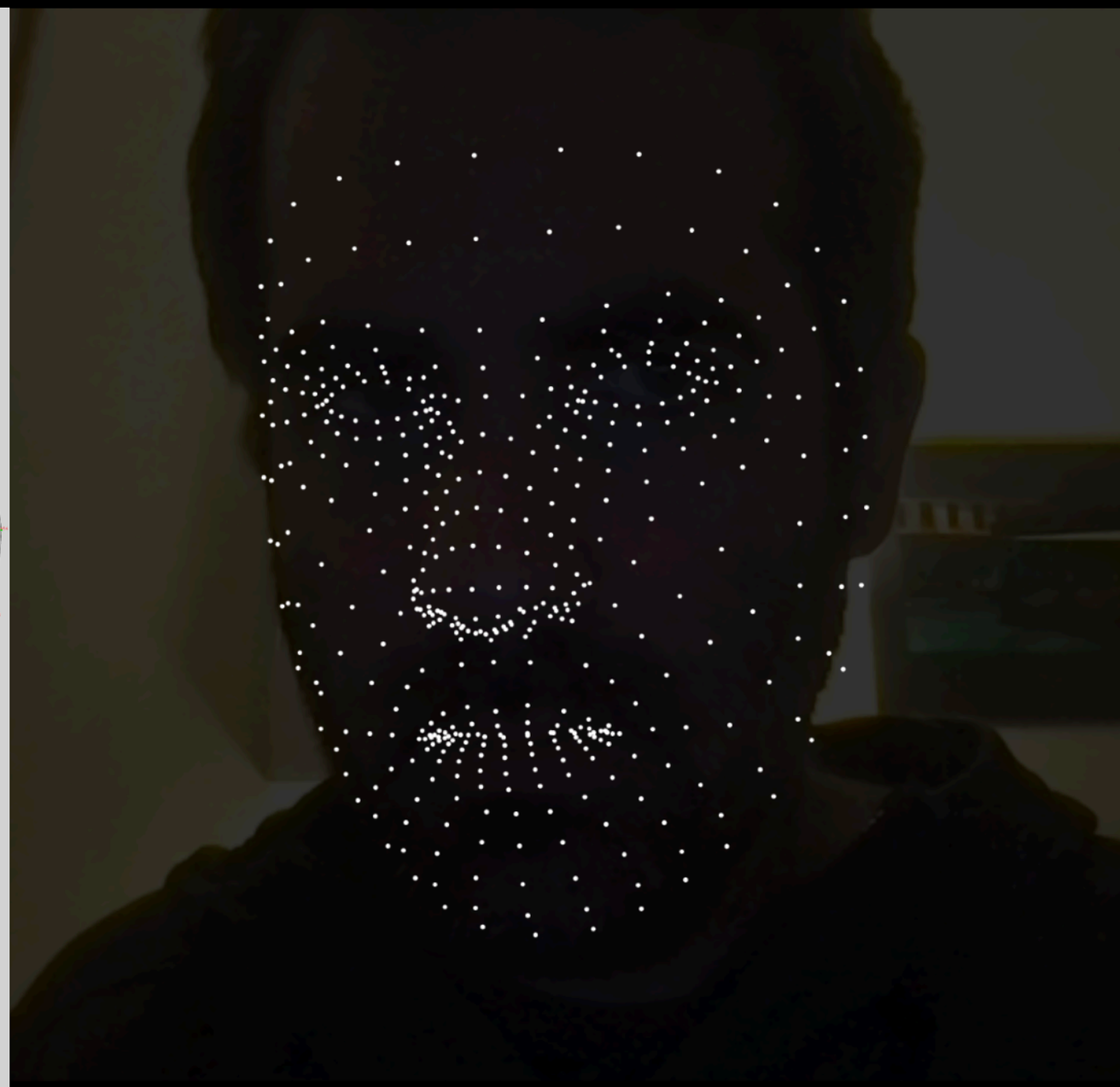
0. Nose
1. Left_eye_inner
2. Left_eye
3. Left_eye_outer
4. Right_eye_inner
5. Right_eye
6. Right_eye_outer
7. Left_ear
8. Right_ear
9. Left_mouth
10. Right_mouth
11. Left_shoulder
12. Right_shoulder
13. Left_elbow
14. Right_elbow
15. Left_wrist
16. Right_wrist
17. Left_pinky
18. Right_pinky
19. Left_index
20. Right_index
21. Left_thumb
22. Right_thumb
23. Left_hip
24. Right_hip
25. Left_knee
26. Right_knee
27. Left_ankle
28. Right_ankle
29. Left_heel
30. Right_heel
31. Left_foot_index
32. Right_foot_index

HANDPOSE : Machine-learning model for hand & finger tracking



FACEMESH : Machine-learning model for facial landmark detection





HANDPOSE : starting from ml5 handpose keypoints example

p5* File Edit Sketch Help English Hello, visiophone!

Auto-refresh 01_Handpose by visiophone

sketch.js • Saved: 1 minute ago

```
14
15  function setup() {
16    createCanvas(640, 480);
17    // Create the webcam video and hide it
18    video = createCapture(VIDEO);
19    video.size(width, height);
20    video.hide();
21
22    // start detecting hands from the webcam video
23    handpose.detectStart(video, gotHands);
24
25  }
26
27  function draw() {
28
29    //////// Flipped webcam
30    translate(width, 0); // move canvas to the right
31    scale(-1.0, 1.0);
32    image(video, 0, 0, width, height); // Draw the webcam video
33    ////////////////
34
35    // Draw all the tracked hand points
36    for (let i = 0; i < hands.length; i++) {
37      let hand = hands[i];
38      for (let j = 0; j < hand.keypoints.length; j++) {
39        let keypoint = hand.keypoints[j];
40        fill(0, 255, 0);
41        noStroke();
42        circle(keypoint.x, keypoint.y, 10);
43      }
44    }
45  }
46  // Callback function for when handpose outputs data
47  function gotHands(results) {
48    // save the output to the hands variable
49    hands = results;
50  }
51
```

Preview

Mirroring the image

Loops through the hands' keypoints and draws a circle on each point

HANDPOSE : lines connecting keypoints

p5* File Edit Sketch Help English Hello, visiophone!

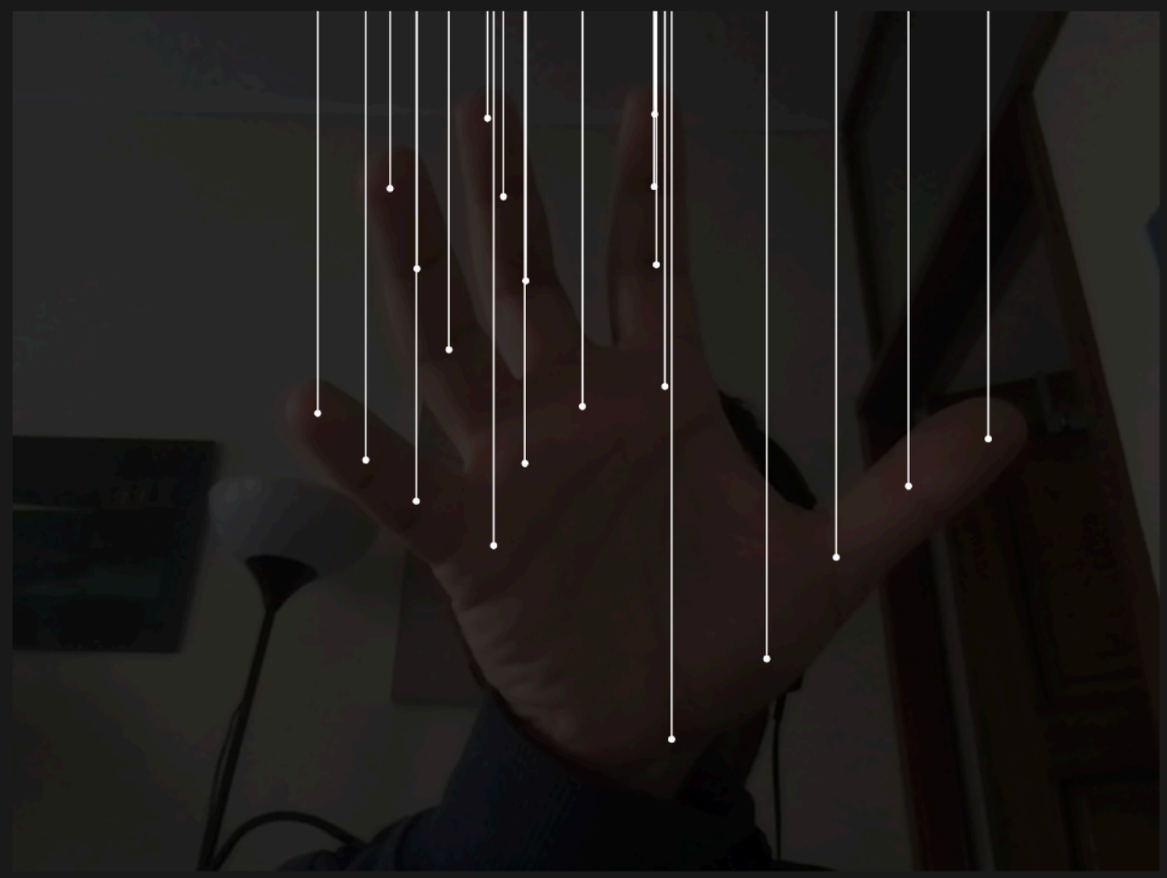
Auto-refresh 01_Handpose_LinesConnecting_Keypoints by visiophone

sketch.js Saved: 2 minutes ago

```
  video.size(width, height);
  video.hide();

  // start detecting hands from the webcam video
  handpose.detectStart(video, gotHands);

}

function draw() {

  //////// Flipped webcam
  translate(width, 0); // move canvas to the right
  scale(-1.0, 1.0);
  image(video, 0, 0, width, height); // Draw the webcam video
 // image(video, width-width/4, 0, width/4, height/4); //video feed in the screen corner
  fill(0,200);noStroke(); // darkening Keypoints
  rect(0,0,width,height);
  ////////////////

  // Draw all the tracked hand points
  for (let i = 0; i < hands.length; i++) {
    let hand = hands[i];

    for (let j = 0; j < hand.keypoints.length; j++) {
      let keypoint = hand.keypoints[j];
      fill(255);
      noStroke();
      circle(keypoint.x, keypoint.y, 4);
      stroke(255);
      // line connecting each point to the top of screen
      line(keypoint.x, keypoint.y,keypoint.x, 0);
    }
  }

/*
  // If we have both hands on screen. Connect lines between L & R hand
  if(hands.length==2){
```

Console

Preview



Lines connecting each point to the top of screen

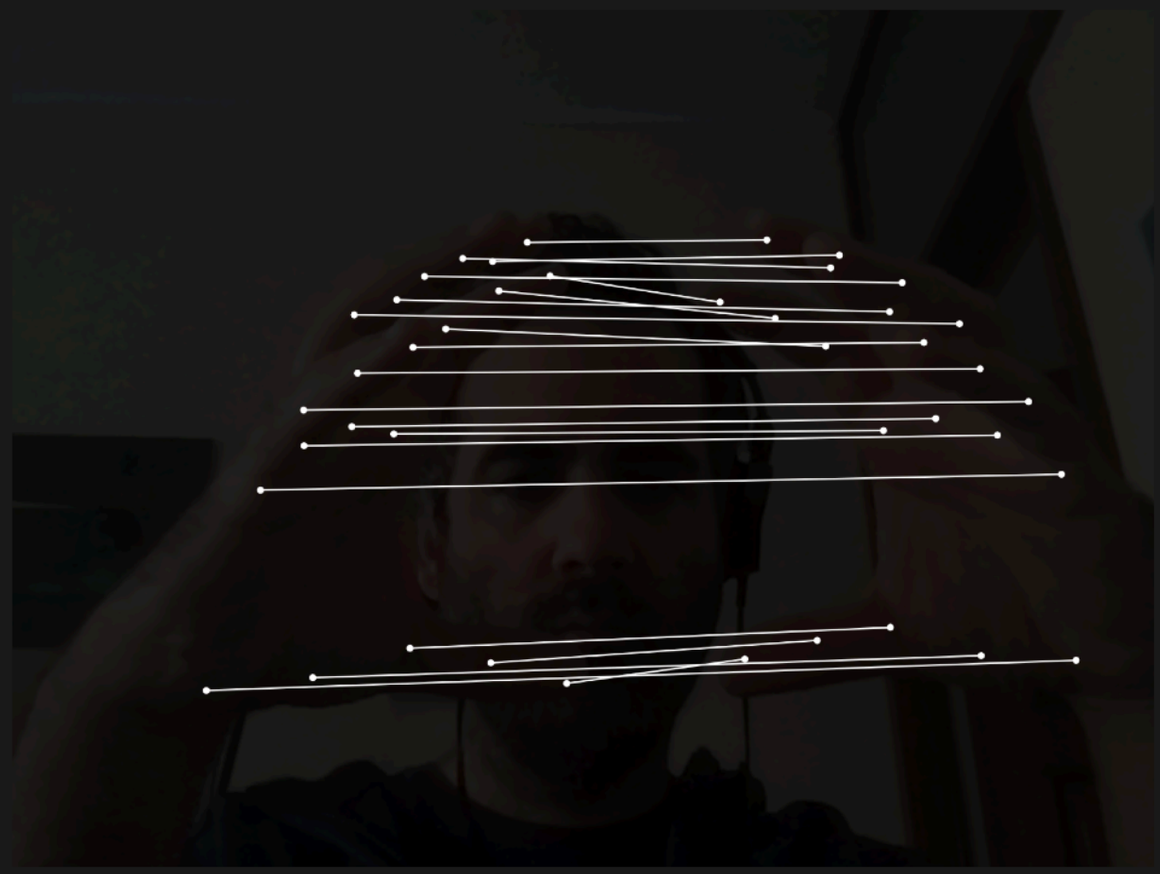
HANDPOSE : lines connecting keypoints

Auto-refresh 01_Handpose_LinesConnecting_Keypoints by visiophone

sketch.js Saved: 25 seconds ago Preview

```
47        circle(keypoint.x, keypoint.y, 4);
48        stroke(255);
49        // line connecting each point to the top of screen
50        line(keypoint.x, keypoint.y,keypoint.x, 0);
51      }
52    }
53  */
54
55    // If we have both hands on screen. Connect lines between L & R hand
56    if(hands.length==2){
57
58      let rHand = hands[0];
59      let lHand = hands[1];
60
61      for (let j = 0; j < rHand.keypoints.length; j++) {
62        let rPoints = rHand.keypoints[j];
63        let lPoints = lHand.keypoints[j];
64
65        fill(255);noStroke();
66        circle(rPoints.x, rPoints.y, 4);
67        circle(lPoints.x, lPoints.y, 4);
68
69        stroke(255);
70        line(rPoints.x, rPoints.y,lPoints.x, lPoints.y);
71
72      }
73
74    }
75
76
77  }
78  // Callback function for when handpose outputs data
79  function gotHands(results) {
80    // save the output to the hands variable
81    hands = results;
82
83  }
84
```

Console



Check if we have two hands on screen

Lines connecting one hand to the other

HANDPOSE : hands data

21 Keypoints (x,y)
21 Keypoints3D (x,y,z)

1 Hand detected

Independent Hand parts
2D & 3D points

sketch.js •

Saved: 7 minutes ago

```
55  // Callback function for when handpose outputs data
56  function gotHands(results) {
57    // save the output to the hands variable
58    hands = results;
59    console.log(hands);
60  }
61
```

Console

Clear

```
▼ (1) [Object]
  ▼ 0: Object
    ▶ keypoints: Array(21)
    ▶ keypoints3D: Array(21)
      score: 0.961761474609375
      handedness: "Right"
    ▼ wrist: Object
        x: 310.5135917663574
        y: 532.8429794311523
        x3D: 0.017497681081295013
        y3D: 0.0825820192694664
        z3D: 0.056304931640625
    ▶ thumb_cmc: Object
    ▶ thumb_mcp: Object
    ▶ thumb_ip: Object
    ▶ thumb_tip: Object
    ▶ index_finger_mcp: Object
    ▶ index_finger_pip: Object
    ▶ index_finger_dip: Object
    ▶ index_finger_tip: Object
    ▶ middle_finger_mcp: Object
    ▶ middle_finger_pip: Object
    ▶ middle_finger_dip: Object
    ▶ middle_finger_tip: Object
    ▶ ring_finger_mcp: Object
    ▶ ring_finger_pip: Object
    ▶ ring_finger_dip: Object
    ▶ ring_finger_tip: Object
    ▶ pinky_finger_mcp: Object
    ▶ pinky_finger_pip: Object
    ▶ pinky_finger_dip: Object
    ▶ pinky_finger_tip: Object
```

HANDPOSE : fingers ID

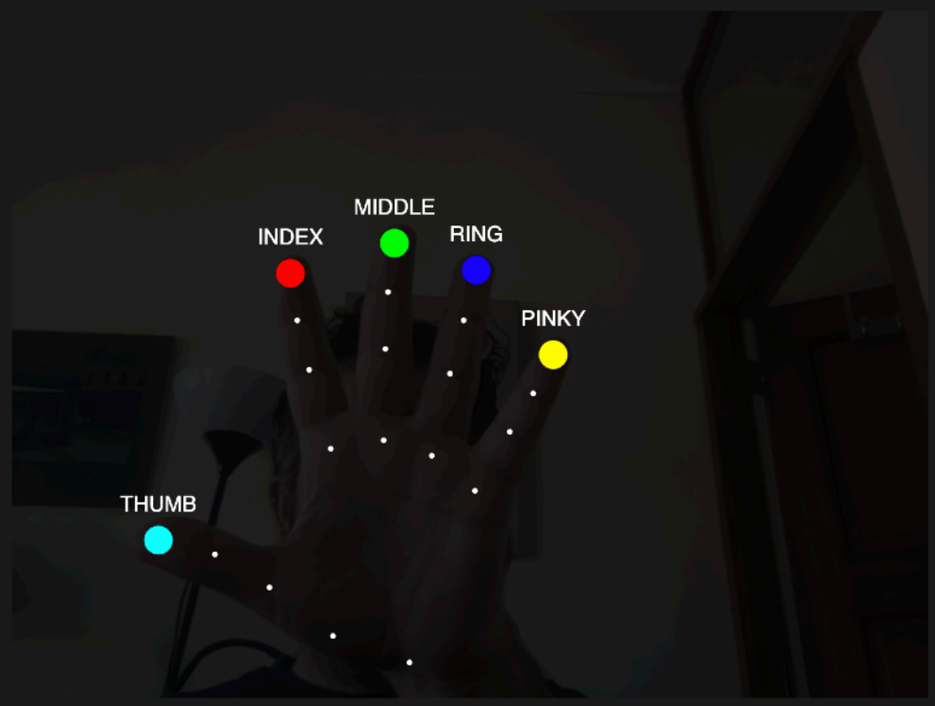
5 fingers tips + wrist

Auto-refresh 02_Handpose_ by visiophone

sketch.js Saved: 15 seconds ago

```
49  ////////////////////////////////////////////////////////////
50
51      // If there is at least one hand
52  if (hands.length > 0) {
53    // Find the index finger tip and thumb tip
54    let h0 = hands[0].wrist;
55    let h1 = hands[0].index_finger_tip;
56    let h2 = hands[0].middle_finger_tip;
57    let h3 = hands[0].ring_finger_tip;
58    let h4 = hands[0].pinky_finger_tip;
59    let h5 = hands[0].thumb_tip;
60
61    // OPEN DIST>0.17
62    print(dist(h0.x3D,h0.y3D,h0.z3D,h5.x3D,h5.y3D,h5.z3D));
63
64    // INDEX UP
65    if(dist(h0.x3D,h0.y3D,h0.z3D,h1.x3D,h1.y3D,h1.z3D)>0.17){
66      fill(255,0,0);noStroke();
67      circle(h1.x,h1.y,20);
68
69      push();
70      scale(-1.0, 1.0);
71      translate(-width, 0); // move canvas to the right
72      textSize(15);textAlign(CENTER);fill(255);
73      text("INDEX",width-h1.x,h1.y-20);
74      pop();
75
76
77  }
78
79  //  MIDDLE UP
80  if(dist(h0.x3D,h0.y3D,h0.z3D,h2.x3D,h2.y3D,h2.z3D)>0.18){
81    fill(0,255,0);noStroke();
82    circle(h2.x,h2.y,20);
83
84    push();
85    scale(-1.0, 1.0);
86    translate(-width, 0); // move canvas to the right
87    textSize(15);textAlign(CENTER);fill(255);
88    text("MIDDLE",width-h2.x,h2.y-20);
89    pop();
90  }
91  //  RING UP
92    if(dist(h0.x3D,h0.y3D,h0.z3D,h3.x3D,h3.y3D,h3.z3D)>0.17){
```

Preview



Checking if finger is UP or DOWN: distance between tip and wrist. If the distance is bigger that 0.17 -> FINGER UP

HANDPOSE : ml5 handpose parts example

Calling specific finger parts (tip of index & thumb)

Auto-refresh 00_Example_ml5-handpose-parts by visiophone

sketch.js Saved: about 4 hours ago

```
let hands = [];

// A variable to track a pinch between thumb and index
let pinch = 0;

function preload() {
  // Load the handpose model.
  handpose = ml5.handpose();
}

function setup() {
  createCanvas(640, 480);
  // Create the webcam video and hide it
  video = createCapture(VIDEO);
  video.size(width, height);
  video.hide();
  // start detecting hands from the webcam video
  handpose.detectStart(video, gotHands);
}

function draw() {
  // Draw the webcam video
  image(video, 0, 0, width, height);

  // If there is at least one hand
  if (hands.length > 0) {
    // Find the index finger tip and thumb tip
    let finger = hands[0].index_finger_tip;
    let thumb = hands[0].thumb_tip;

    // Draw circles at finger positions
    let centerX = (finger.x + thumb.x) / 2;
    let centerY = (finger.y + thumb.y) / 2;
    // Calculate the pinch "distance" between finger and thumb
    let pinch = dist(finger.x, finger.y, thumb.x, thumb.y);

    // This circle's size is controlled by a "pinch" gesture
    fill(0, 255, 0, 200);
    stroke(0);
    strokeWeight(2);
    circle(centerX, centerY, pinch);
  }
}

// Callback function for when handpose outputs data
function gotHands(results) {
  // save the output to the hands variable
  hands = results;
  // console.log(hands);
}
```

Preview

Distance between booth fingers

Finding the middle point between booth fingers

HANDPOSE : Drawing with the fingers (erase_background)

Webcam video only in the corner of the screen. No background color

Auto-refresh 03_Handpose_fingerDrawing_eraseBackground by visiophone

sketch.js Saved: 25 seconds ago

```
  ////// Flipped webcam
  translate(width, 0); // move canvas to the right
  scale(-1.0, 1.0);
 // image(video, 0, 0, width, height); // Draw the webcam video
  image(video, width-width/4, 0, width/4, height/4); //video feed in the
screen corner
  fill(0,220);noStroke(); // darkening Keypoints
  //rect(0,0,width,height);
  ////////////////

////////////////////////////////////////////////////////////////////
  // Draw all the tracked hand points // circle in each part
  for (let i = 0; i < hands.length; i++) {
    let hand = hands[i];
    for (let j = 0; j < hand.keypoints.length; j++) {
      let keypoint = hand.keypoints[j];
      fill(255);
      noStroke();
    //  circle(keypoint.x, keypoint.y, 4);
      stroke(255);
    }
  ////////////////////////////////////////////////////////////////////

    ///
 // If there is at least one hand
  if (hands.length > 0) {
    // Find the index finger tip and thumb tip
    let finger = hands[0].index_finger_tip;
    let thumb = hands[0].thumb_tip;

    // Draw circles at finger positions
    let centerX = (finger.x + thumb.x) / 2;
    let centerY = (finger.y + thumb.y) / 2;
    // Calculate the pinch "distance" between finger and thumb
    let pinch = dist(finger.x, finger.y, thumb.x, thumb.y);

    // This circle's size is controlled by a "pinch" gesture
    fill(255);
    stroke(0);
    strokeWeight(1);
    circle(centerX, centerY, pinch);
  }
```

Preview

Drawing circles

HANDPOSE : Drawing with the fingers (saving points on array)

```
12
13  let polyline = []; // array to store the drawing points
14  let wLine = [];  // array to store line's weight
15
```

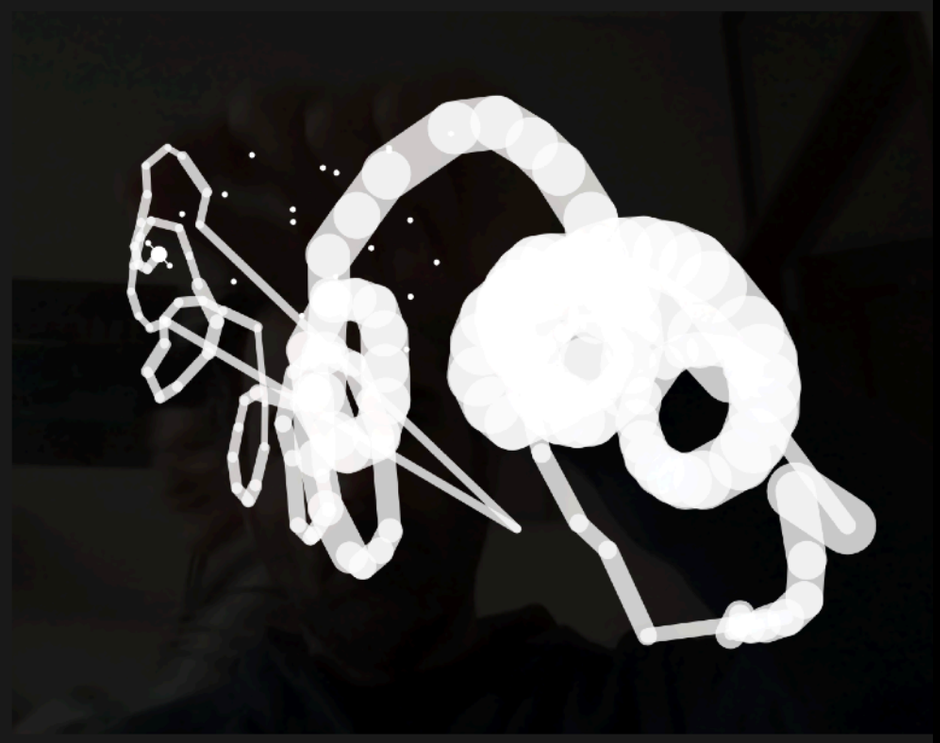
← Adding on the top

Auto-refresh 04_Handpose_fingersDrawing by visiophone

sketch.js Saved: 2 minutes ago Preview

```
54
55    // If there is at least one hand
56    if (hands.length > 0) {
57      // Find the index finger tip and thumb tip
58      let finger = hands[0].index_finger_tip;
59      let thumb = hands[0].thumb_tip;
60
61      // Draw circles at finger positions
62      let centerX = (finger.x + thumb.x) / 2;
63      let centerY = (finger.y + thumb.y) / 2;
64      // Calculate the pinch "distance" between finger and thumb
65      let pinch = dist(finger.x, finger.y, thumb.x, thumb.y);
66      strokeWeight(1);
67      line(finger.x,finger.y,thumb.x,thumb.y); circle(centerX,centerY,10);
68
69      // when we have hands on the screen add positions to the polynie array
70      polyline.push(createVector(centerX, centerY));
71      wLine.push(pinch/4);
72    //strokeWeight(pinch);
73
74    }
75     ////////////////////////////////////////////////////////////////
76      // draw lines according the size of polyline array
77      for (let i = 0; i < polyline.length; i++) {
78
79        // for all elements except for the last one
80        if (i + 1 < polyline.length) {
81          // draw a line between current element and the next one
82          stroke(255,200);
83          strokeWeight(wLine[i]);
84          line(polyline[i].x, polyline[i].y, polyline[i+1].x, polyline[i+1].y);
85        }
86      }
87
88
89
90  }
91  // Callback function for when handpose outputs data
92  function gotHands(results) {
93    // save the output to the hands variable
94    hands = results;
95    //console.log(hands);
96  }
97
```

← Saving Hands position



↑ Drawing the line

HANDPOSE : Drawing with the fingers (saving points on array)

Smoothing the position values, so the drawing is more fluid

Auto-refresh 04_Handpose_fingersDrawing by visiophone

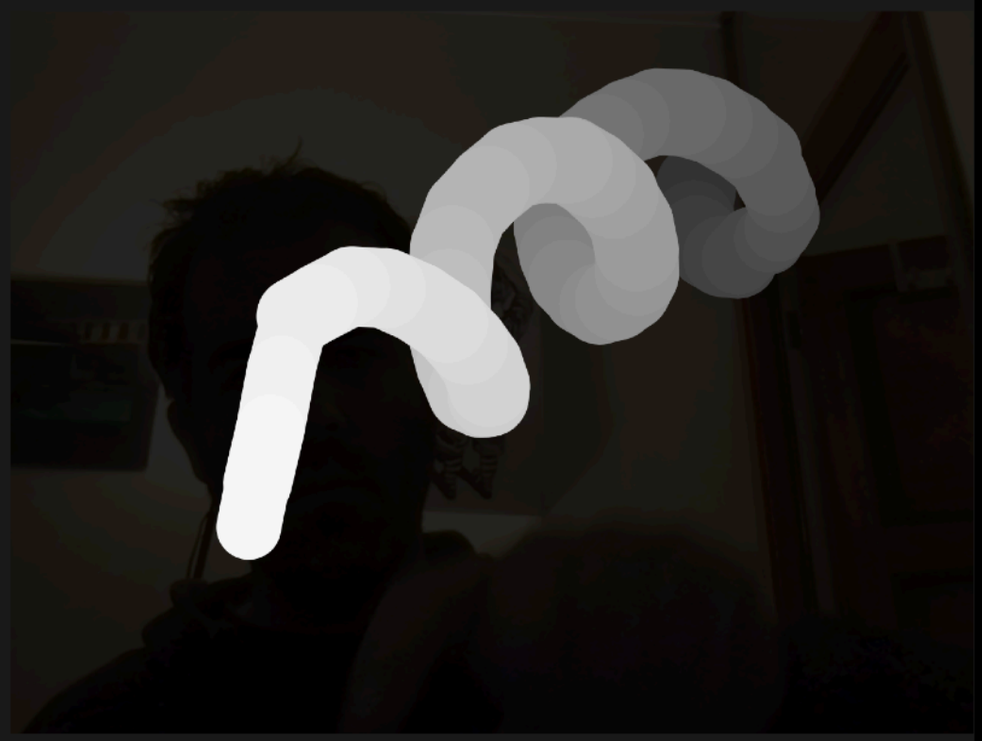
Stop sketch

sketch.js

Saved: 3 minutes ago

Preview

```
59  // if there is at least one hand
60  if (hands.length > 0) {
61    // Find the index finger tip and thumb tip
62    let finger = hands[0].index_finger_tip;
63    let thumb = hands[0].thumb_tip;
64
65    // Draw circles at finger positions
66    let centerX = (finger.x + thumb.x) / 2;
67    let centerY = (finger.y + thumb.y) / 2;
68    // Calculate the pinch "distance" between finger and thumb
69    let pinch = dist(finger.x, finger.y, thumb.x, thumb.y);
70
71    // smoothing values
72    posX+=(centerX-posX)*0.2;
73    posY+=(centerY-posY)*0.2;
74    wLinee+=(pinch/2-wLinee)*0.2;
75
76    strokeWeight(1); // line connecting the fingers
77    line(finger.x,finger.y,posX,posY); line(thumb.x,thumb.y,posX,posY);
78
79    // when we have hands on the screen add positions to the polynie array
80    polyline.push(createVector(posX, posY));
81    wLine.push(wLinee);
82
83
84  }
85   ////////////////////////////////////////////////////////////////
86    // draw lines according the size of polyline array
87    for (let i = 0; i < polyline.length; i++) {
88
89      // for all elements except for the last one
90      if (i + 1 < polyline.length) {
91        // draw a line between current element and the next one
92        stroke(map(i,0,polyline.length,50,255));
93        strokeWeight(wLine[i]);
94        line(polyline[i].x, polyline[i].y, polyline[i+1].x, polyline[i+1].y);
95      }
96    }
97
98
99  // Settingn a limite to the site of the line
100 if(polyline.length>60) {polyline.shift();wLine.shift();}
101
102
103
```



Line Size limite. When it reaches the limit the first point is erased

BLAZEPOSE : starting from ml5 Blazepose keypoints example

Auto-refresh 05_Blazepose by visiophone

Play sketch

sketch.js • Saved: 1 minute ago

Preview

```
Based on the examples by ml5 (https://editor.p5js.org/ml5/sketches/OukJYAJAb)
*/

let video;
let bodypose;
let poses = [];

function preload() {
  //Load the bodypose model.
  bodypose = ml5.bodypose("BlazePose");
}

function setup() {
  createCanvas(640, 480);

  // Create the video and hide it
  video = createCapture(VIDEO);
  video.size(width, height);
  video.hide();

  // Start detecting poses in the webcam video
  bodypose.detectStart(video, gotPoses);
}

function draw() {
  //////// Flipped webcam
  translate(width, 0); // move canvas to the right
  scale(-1.0, 1.0);
   image(video, 0, 0, width, height); // Draw the webcam video
  //image(video, width - width / 4, 0, width / 4, height / 4); //video feed in the screen corner
  fill(0, 220);
  noStroke(); // darkening Keypoints
  rect(0,0,width,height);
  ////////////////////////////////////////////////////////////

  // Draw all the tracked landmark points
  for (let i = 0; i < poses.length; i++) {
    let pose = poses[i];
    for (let j = 0; j < pose.keypoints.length; j++) {
      let keypoint = pose.keypoints[j];
      fill(0, 255, 0);
      noStroke();
      circle(keypoint.x, keypoint.y, 10);
```

Mirroring & darkening the image

Loops through the body' keypoints and draws a circle on each point

BLAZEPOSE : body data

21 Keypoints (x,y)
21 Keypoints3D (x,y,z)

1 Body detected

Independent Body parts
2D & 3D points

Console

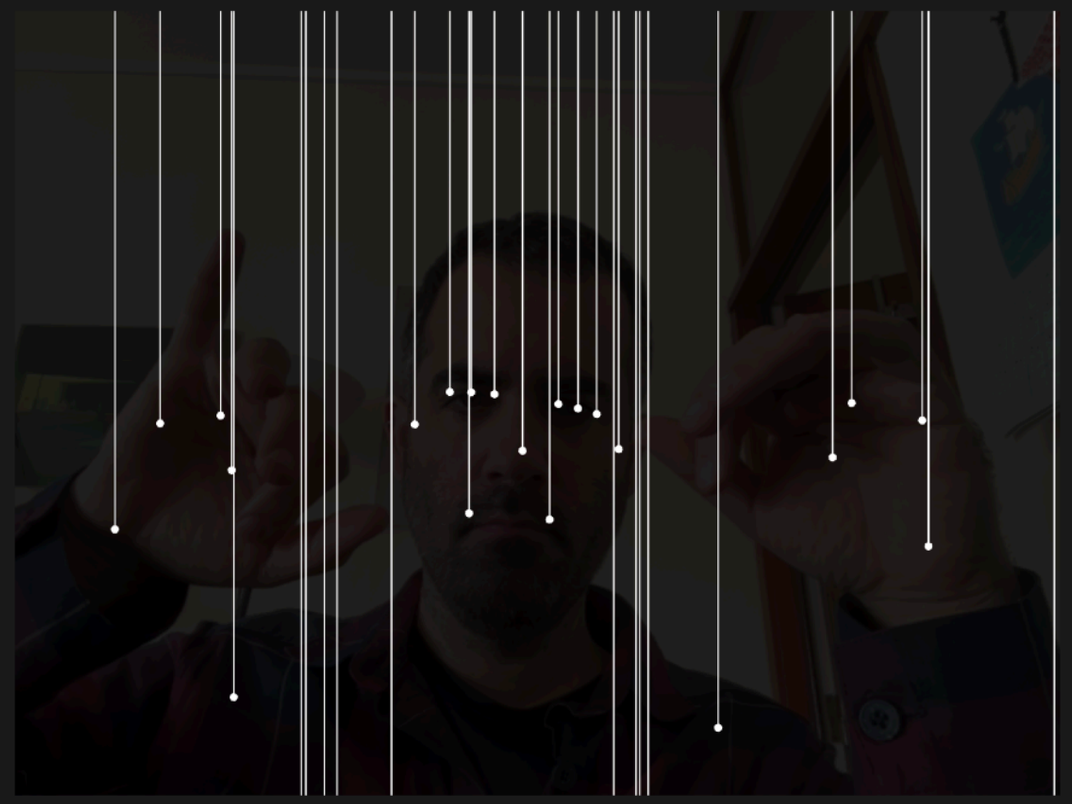
▼0: Object
▶keypoints: Array(33)
▶keypoints3D: Array(33)
▶nose: Object
▶left_eye_inner: Object
▶left_eye: Object
▶left_eye_outer: Object
▶right_eye_inner: Object
▶right_eye: Object
▶right_eye_outer: Object
▶left_ear: Object
▶right_ear: Object
▶mouth_left: Object
▶mouth_right: Object
▶left_shoulder: Object
▶right_shoulder: Object
▶left_elbow: Object
▶right_elbow: Object
▶left_wrist: Object
▶right_wrist: Object
▶left_pinky: Object
▶right_pinky: Object
▶left_index: Object
▶right_index: Object
▶left_thumb: Object
▶right_thumb: Object
▶left_hip: Object
▶right_hip: Object
▶left_knee: Object
▶right_knee: Object
▶left_ankle: Object
▶right_ankle: Object
▶left_heel: Object
▶right_heel: Object
▶left_foot_index: Object
▶right_foot_index: Object

BLAZEPOSE : lines connecting keypoints

Auto-refresh 05_BlazePose_keypoints_Lines by visiophone

sketch.js Saved: 3 minutes ago Preview

```
}

function draw() {
    //////// Flipped webcam
  translate(width, 0); // move canvas to the right
  scale(-1.0, 1.0);
   image(video, 0, 0, width, height); // Draw the webcam video
  //image(video, width - width / 4, 0, width / 4, height / 4);
//video feed in the screen corner
  fill(0, 220);
  noStroke(); // darkening Keypoints
  rect(0,0,width,height);
  ////////////////////////////////////////////////////////////////////

  // Draw all the tracked landmark points
  for (let i = 0; i < poses.length; i++) {
    let pose = poses[i];
    for (let j = 0; j < pose.keypoints.length; j++) {
      let keypoint = pose.keypoints[j];
      fill(255);
      noStroke();
      circle(keypoint.x, keypoint.y, 5);
      stroke(255);
      line(keypoint.x, keypoint.y,keypoint.x, -100);
    }
  }
}
```



Lines connecting each point to the top of screen

BLAZEPOSE : eyes

Calling specific body parts (right & left eyes)

Auto-refresh 06_Blazepose_eyes by visiophone

sketch.js Saved: 14 days ago Preview

```
48
49 if (poses.length > 0) {
50
51   //smooth values to drag the inner eyes circle
52   leftX+=(poses[0].left_eye.x-leftX)*0.1;
53   leftY+=(poses[0].left_eye.y-leftY)*0.1;
54
55   rightX+=(poses[0].right_eye.x-rightX)*0 1;
56   rightY+=(poses[0].right_eye.y-rightY)*0 1;
57
58   stroke(255);
59   //line connecting realtime eyes position to the smoother position
60   line(poses[0].left_eye.x,poses[0].left_eye.y,leftX,leftY);
61   line(poses[0].right_eye.x,poses[0].right_eye.y,rightX,rightY);
62
63   fill(255);
64
65   circle(poses[0].left_eye.x,poses[0].left_eye.y,60); // eye's
   circle
66   circle(poses[0].right_eye.x,poses[0].right_eye.y,60);
67   fill(0);
68   circle(leftX,leftY,40); // inner eyes circle
69   circle(rightX,rightY,40);
70
71 }
72
73 }
74
75 // Callback function for when bodypose outputs data
```

Drawing a circle on the eyes’ position

BLAZEPOSE : mask

Fixed randomSeed for deterministic randomness

Calling specific body parts (right & left eyes, nouse, mouth)

Auto-refresh 07_Blazepose_Mask by visiophone

sketch.js Saved: 14 days ago Preview

```
48
49    randomSeed(seed); // fix the random number
50
51  if (poses.length > 0) {
52
53    fill(255); stroke(0);strokeWeight(1);
54    // Eyes External circle. Circle size is random
55    leftEye=int(random(15,150));
56    rightEye=int(random(15,150));
57    circle(poses[0].left_eye.x,poses[0].left_eye.y,leftEye);
58
      circle(poses[0].right_eye.x,poses[0].right_eye.y,rightEye
      );
59    // Eyes Inner circle
60    fill(0);
61     circle(poses[0].left_eye.x,poses[0].left_eye.y,10);
62    circle(poses[0].right_eye.x,poses[0].right_eye.y,10);
63    // mouth
64    stroke(255); strokeCap(SQUARE);
65    mouth=int(random(1,30));
66    strokeWeight(mouth);
67
      line(poses[0].mouth_left.x,poses[0].mouth_left.y,poses[0]
      .mouth_right.x,poses[0].mouth_right.y);
68      // nose
69    fill(255,0,0);noStroke();
70    nose=int(random(15,70));
71    circle(poses[0].nose.x,poses[0].nose.y,nose);
72
```

Drawing circles and lines with random sizes

```
84
85  function mouseReleased() {
86
87    seed=int(random(millis()));
88    print("RandomSeed: "+seed);
89  }
90
```

Mouse click changes randomSeed

BLAZEPOSE : mask PNG

Places images (png with no background) on specific body parts

Auto-refresh 08_Blazepose_Mask_pngs by visiophone

sketch.js Saved: 14 days ago Preview

```
49  ////////////////////////////////////////////////////////////////
50    randomSeed(seed); // fix the random number
51
52  if (poses.length > 0) {
53    imageMode(CENTER); // position image from the center
54
55    randLeft=int(random(5)); // randomize image
56    randRight=int(random(5));
57    leftEye=int(random(15,150)); // randomize size
58    rightEye=int(random(15,150));
59    // insert the random image in the eye's position
60
      image(pic[randLeft],poses[0].left_eye.x,poses[0].left_eye.y,leftEye,leftEye);
61
      image(pic[randRight],poses[0].right_eye.x,poses[0].right_eye.y,rightEye,right
    Eye);
62    imageMode(CORNER); // position image from the corner
63
64  }
65  }
66
67  // Callback function for when bodypose outputs data
68  function gotPoses(results) {
```

Console

```
RandomSeed: 18130
RandomSeed: 11606
RandomSeed: 11580
```

```
21  function preload() {
22    bodypose = ml5.bodypose("BlazePose");  //Load the bodypose model.
23    pic[0] = loadImage('pic/eye0.png');
24    pic[1] = loadImage('pic/eye1.png');
25    pic[2] = loadImage('pic/eye2.png');
26    pic[3] = loadImage('pic/eye3.png');
27    pic[4] = loadImage('pic/eye4.png');
28  }
29
```

Image number and size are randomised

Images are pre-loaded into pic[] array

FACEMESH : ml5 facemesh keypoints example

Auto-refresh 00_Example_ml5-facemesh_keypoints by visiophone

sketch.js Saved: about 21 hours ago

```
  // Create the webcam video and hide it
  video = createCapture(VIDEO);
  video.size(width, height);
  video.hide();
  // Start detecting faces from the webcam video
  facemesh.detectStart(video, gotFaces);
}

function draw() {
  // Draw the webcam video
  image(video, 0, 0, width, height);

  // Draw all the tracked face points
  for (let i = 0; i < faces.length; i++) {
    let face = faces[i];
    for (let j = 0; j < face.keypoints.length; j++) {
      let keypoint = face.keypoints[j];
      fill(0, 255, 0);
      noStroke();
      circle(keypoint.x, keypoint.y, 5);
    }
  }
}
```

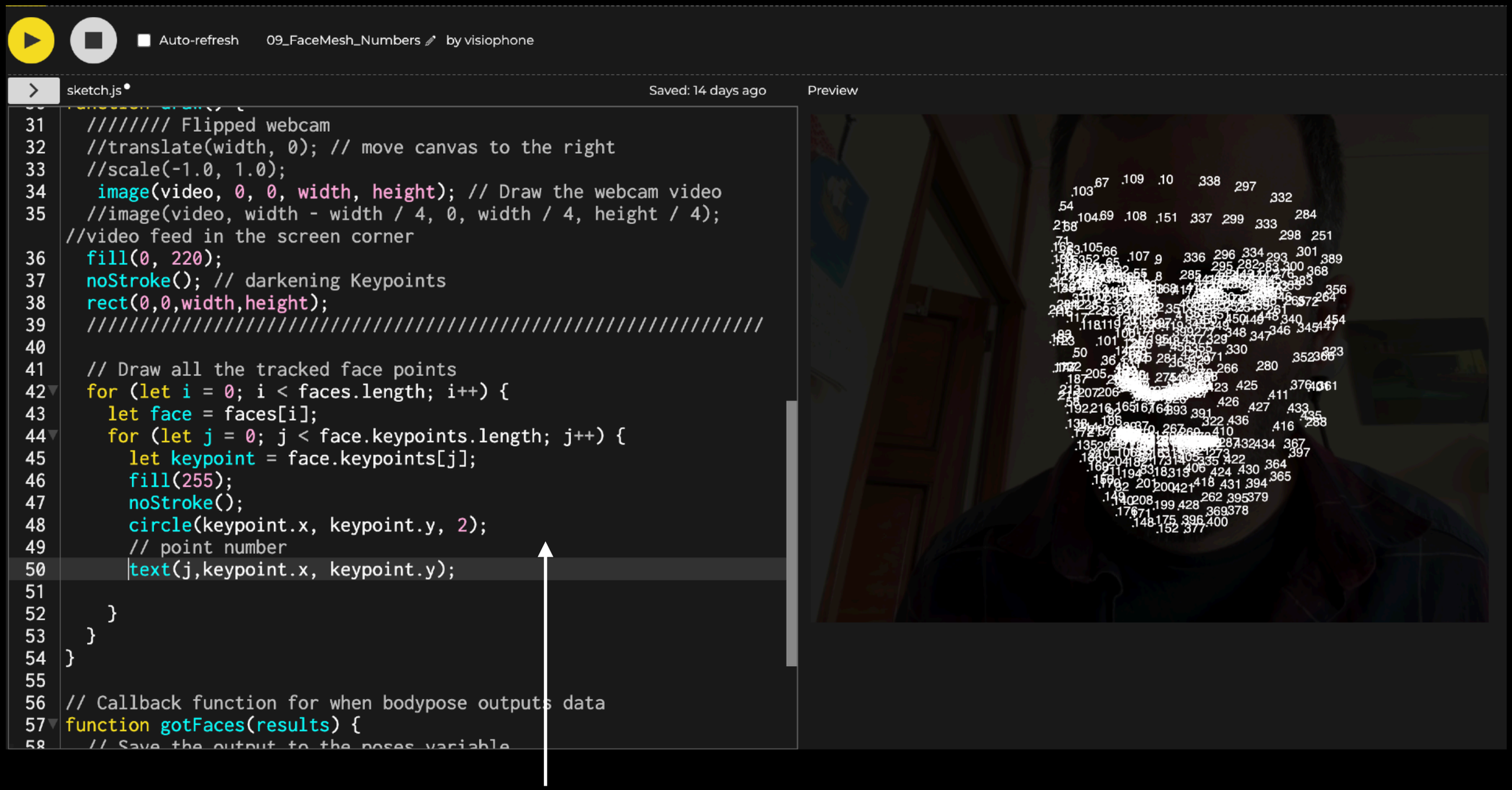
Console Clear

```
(OpenGL ES 3.0 (WebGL 2.0 (OpenGL ES 3.0 Chromium)))
W0221 15:02:13.597000 1 gl_context.cc:821] OpenGL error chec
king is disabled
```

Preview

Loops through face’s keypoints and draws a circle on each point

FACEMESH : points numbers ID



Loops through face’s keypoints and writes point number ID

FACEMESH : FaceData

1 Face detected

458 Keypoints (x,y,z)

Independent Face parts
2D & 3D points

Console

```
▼ (1) [Object]
  ▼ 0: Object
    ▶ keypoints: Array(468)
    ▶ box: Object
    ▶ lips: Array(39)
    ▶ rightEye: Array(16)
    ▶ faceOval: Array(36)
    ▶ rightEyebrow: Array(9)
    ▶ leftEye: Array(16)
    ▼ leftEyebrow: Array(9)
      ▼ 0: Object
          x: 434.49268341064453
          y: 168.61254215240479
          z: 16.16122841835022
      ▼ 1: Object
          x: 419.95471954345703
          y: 160.27111530303955
          z: 1.0547345131635666
      ▼ 2: Object
          x: 428.66199493408203
          y: 163.43931198120117
          z: 7.664891481399536
      ▶ 3: Object
      ▶ 4: Object
```

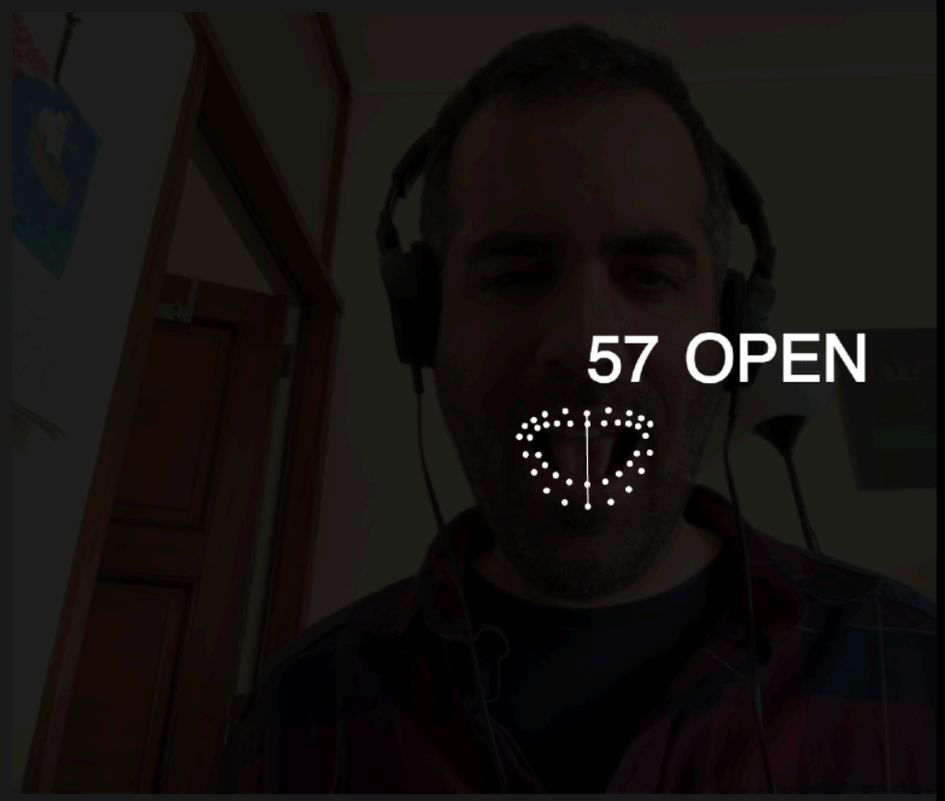
FACEMESH : mouth points

Calling specific body parts (lips)

Auto-refresh 10_FaceMesh_mouth by visiophone

sketch.js Saved: 14 days ago Preview

```
in the screen corner
  fill(0, 220);
  noStroke(); // darkening Keypoints
  rect(0,0,width,height);
  ////////////////////////////////////////////////////////////

   if (faces.length > 0) {
for (let i = 0; i < faces[0].lips.length; i++) {
      let lips = faces[0].lips[i];
      fill( 255);noStroke();
      circle(lips.x, lips.y, 4);
      //text(i,lips.x, lips.y-5);
    }
     let topLip = faces[0].lips[0];
     let lowLip =  faces[0].lips[3];

     stroke(255);
     line(topLip.x,topLip.y,lowLip.x,lowLip.y);
     let distt=int(dist(topLip.x,topLip.y,lowLip.x,lowLip.y));
     textSize(40);
     text(distt,topLip.x,topLip.y-20);
     if(distt>50)text("OPEN",topLip.x+60,topLip.y-20);
     else text("CLOSE",topLip.x+60,topLip.y-20);

   }
}

// Callback function for when bodypose outputs data
function gotFaces(results) {
```



measuring the distance between upper and lower lip points to check if mouth is open or closed

FACEMESH : mouth points

Auto-refresh 11_FaceMesh_mouth _Particles by visiophone

sketch.js Saved: 14 days ago

```
45
46    if (faces.length > 0) {
47  for (let i = 0; i < faces[0].lips.length; i++) {
48        let lips = faces[0].lips[i];
49        fill( 255);noStroke();
50        circle(lips.x, lips.y, 4);
51
52      }
53       // distance between top and bottom mounth points to check if
    mouth is open
54       let topLip = faces[0].lips[0];
55       let lowLip =  faces[0].lips[3];
56       let distt=int(dist(topLip.x,topLip.y,lowLip.x,lowLip.y));
57       // if open push a new particle with random size and vel
58       if(distt>40 && frameCount%2==0){
59         let midPointX=(lowLip.x+topLip.x)/2;
60         let midPointY=(lowLip.y+topLip.y)/2;
61         particles.push(createVector(midPointX,midPointY));
62         sizes.push(random(5,50));
63         vel.push(createVector(random(-0.5,0.5),random(-1.5,-0.3)));
64
65       }
66
67     }
68
69
70   // draw the particles. if the particles crosses the top part of
    the screen it is erased
71   for(let i=0; i<particles.length;i++){
```

Preview

If mouth is open (distance between upper & lower lip bigger than 40) then create particles

FACEMESH : mouth particles

Auto-refresh 11_FaceMesh_mouth _Particles by visiophone

sketch.js Saved: 14 days ago

```
45
46    if (faces.length > 0) {
47  for (let i = 0; i < faces[0].lips.length; i++) {
48        let lips = faces[0].lips[i];
49        fill( 255);noStroke();
50        circle(lips.x, lips.y, 4);
51
52      }
53       // distance between top and bottom mounth points to check if
    mouth is open
54       let topLip = faces[0].lips[0];
55       let lowLip =  faces[0].lips[3];
56       let distt=int(dist(topLip.x,topLip.y,lowLip.x,lowLip.y));
57       // if open push a new particle with random size and vel
58       if(distt>40 && frameCount%2==0){
59         let midPointX=(lowLip.x+topLip.x)/2;
60         let midPointY=(lowLip.y+topLip.y)/2;
61         particles.push(createVector(midPointX,midPointY));
62         sizes.push(random(5,50));
63         vel.push(createVector(random(-0.5,0.5),random(-1.5,-0.3)));
64
65       }
66
67     }
68
69
70   // draw the particles. if the particles crosses the top part of
    the screen it is erased
71   for(let i=0; i<particles.length;i++){
```

Preview

If mouth is open (distance between upper & lower lip bigger than 40) then create particles

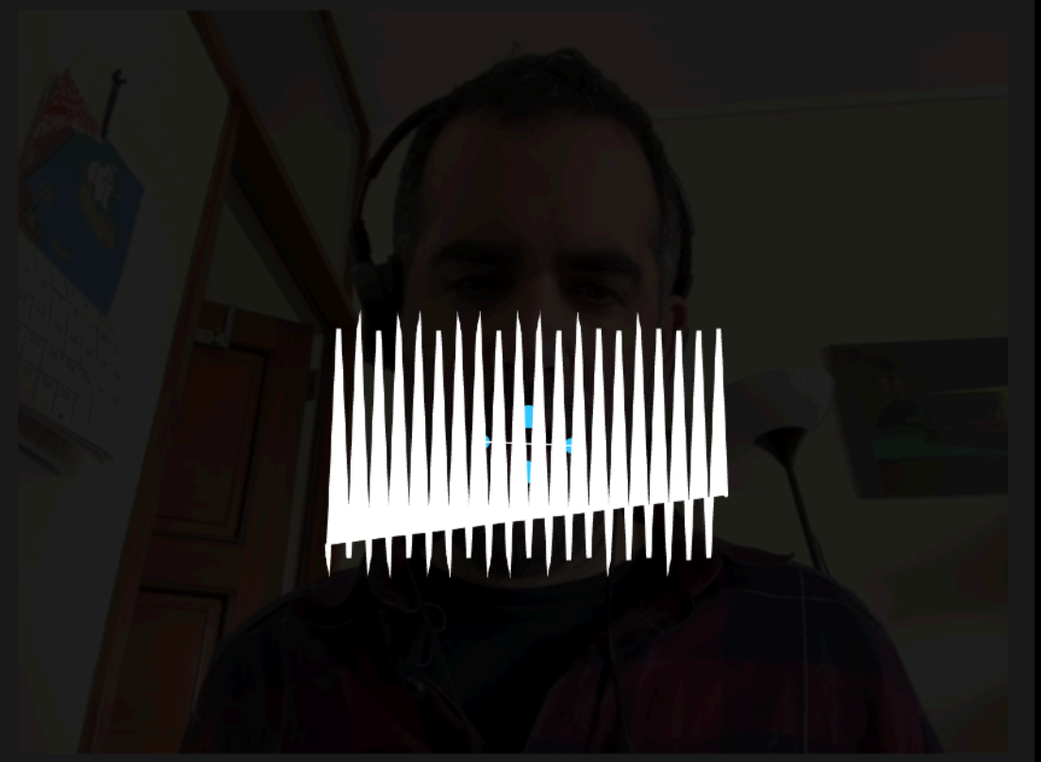
FACEMESH : mouth Oscillator

Mouth Opening changes oscillator frequency

Auto-refresh 12_FaceMesh_mouth _SOUND by visiophone

sketch.js Saved: about 1 hour ago Preview

```
66    let leftLip = faces[0].lips[17];
67    let rightLip = faces[0].lips[34];
68
69    circle(leftLip.x, leftLip.y, 10);
70    circle(rightLip.x, rightLip.y, 10);
71    stroke(255); strokeWeight(1);
72    line(leftLip.x, leftLip.y,rightLip.x, rightLip.y);
73
74    // mouth opening/close changes oscilator's frequency
75    let freqq = map(distt, 20, 70, 20, 2880);
76    osc.freq(freqq);
77
78
79    // DRAW WAVE
80
81
82    fill(255);
83    let waveform = fft.waveform(); // analyze the waveform
84    beginShape();
85    strokeWeight(3);
86    let offSetX=100; //make the wave a little bigger than the mouth
87    for (let i = 0; i < waveform.length; i++) {
88      let x = map(i, 0, waveform.length, leftLip.x-offSetX,
```



Clear

```
351] GL Version: 3.0
omium)))
321] OpenGL error chec
```

Draws audiowave at mouth position

```
13 // Sound
14 let osc, fft;
15
16 function preload() {
17   // Load the facemesh model
18   facemesh = ml5.facemesh();
19 }
20
21 function setup() {
22   createCanvas(640, 480);
23   // Create the webcam video and hide it
24   video = createCapture(VIDEO);
25   video.size(width, height);
26   video.hide();
27   // Start detecting faces from the webcam video
28   facemesh.detectStart(video, gotFaces);
29
30   // SOUND
31   osc = new p5.TriOsc(); // set frequency and type
32   osc.amp(0.5);
33   fft = new p5.FFT();
34   osc.start();
35 }
```

P5Sound Library. Start Oscillator