

Sketch.js

```
let video
let faceMesh
let faces = []
let emitter

function preload()
{
  faceMesh = ml5.faceMesh({ maxFaces: 1, flipped: true })
}

function mousePressed()
{
  console.log(faces)
}

function gotFaces(results)
{
  faces = results
}

function setup()
{
  createCanvas(640, 480)
  video = createCapture(VIDEO, { flipped: true })
  video.hide()
  faceMesh.detectStart(video, gotFaces)
  emitter = new Emitter(300, 300)
}

function draw()
{
  background(0)
  image(video, 0, 0)
  if (faces.length > 0)
  {
    let face = faces[0]
    for (let i = 0; i < face.keypoints.length; i++)
    {
      let keypoint = face.keypoints[i]
      stroke(255, 255, 0)
      strokeWeight(1)
      point(keypoint.x, keypoint.y)
    }
    stroke(255, 255, 0)
    strokeWeight(2)
    beginShape()
    for (let i = 0; i < lipsExterior.length; i++)
    {
      let index = lipsExterior[i]
```

```

    let keypoint = face.keypoints[index]
    noFill()
    vertex(keypoint.x, keypoint.y)
  }
endShape(CLOSE)

beginShape()
for (let i = 0; i < lipsInterior.length; i++)
{
  let index = lipsInterior[i]
  let keypoint = face.keypoints[index]
  noFill()
  vertex(keypoint.x, keypoint.y)
}
endShape(CLOSE)

let a = face.keypoints[13]
let b = face.keypoints[14]
let d = dist(a.x, a.y, b.x, b.y)
let x = (a.x + b.x) * 0.5
let y = (a.y + b.y) * 0.5

let nose = face.keypoints[19]
if (d > 20 && random(1) < 0.25)
{
  emitter.addParticle(x, y, d)
}
}
emitter.run()
}

// Thank you Jack B. Du for these lists!
// Define the exterior lip landmark indices for drawing the outer
lip contour
let lipsExterior = [
  267,
  269,
  270,
  409,
  291,
  375,
  321,
  405,
  314,
  17,
  84,
  181,
  91,
  146,
  61,
  185,
  40,

```

```
    39,  
    37,  
    0,  
]  
  
// Define the interior lip landmark indices for drawing the inner  
lip contour  
let lipsInterior = [  
    13,  
    312,  
    311,  
    310,  
    415,  
    308,  
    324,  
    318,  
    402,  
    317,  
    14,  
    87,  
    178,  
    88,  
    95,  
    78,  
    191,  
    80,  
    81,  
    82,  
]
```

Emitter.js

```
class Emitter
{
  constructor(x, y)
  {
    this.origin = createVector(x, y)
    this.particles = []
  }

  addParticle()
  {
    this.particles.push(new Particle(this.origin.x,
this.origin.y))
  }

  addParticle(x, y, d)
  {
    this.particles.push(new Particle(x, y, d))
  }

  run()
  {
    // Looping through backwards to delete
    for (let i = this.particles.length - 1; i >= 0; i--)
    {
      this.particles[i].run()
      if (this.particles[i].isDead())
      {
        // Remove the particle
        this.particles.splice(i, 1)
      }
    }
  }
}
```

Particle.js

```
class Particle
{
  constructor(x, y, d)
  {
    this.position = createVector(x, y)
    this.acceleration = createVector(0, 0)
    this.velocity = p5.Vector.random2D()
    //this.velocity.mult(random(2, 5))
    this.lifespan = 255.0
    this.r = d * 0.5
  }

  run()
  {
    let gravity = createVector(0, -0.05)
    this.applyForce(gravity)
    this.update()
    this.show()
  }

  applyForce(force)
  {
    this.acceleration.add(force)
  }

  // Method to update position
  update()
  {
    this.velocity.add(this.acceleration)
    this.position.add(this.velocity)
    this.lifespan -= 2
    this.acceleration.mult(0)
  }

  // Method to display
  show()
  {
    stroke(255, this.lifespan)
    fill(255, this.lifespan * 0.5)
    strokeWeight(4)
    circle(this.position.x, this.position.y, this.r * 2)
  }

  // Is the particle still useful?
  isDead()
  {
    return this.lifespan < 0.0
  }
}
```