Algorithmic Art

The Joy of Coding

Module G Unit #3

faceMesh

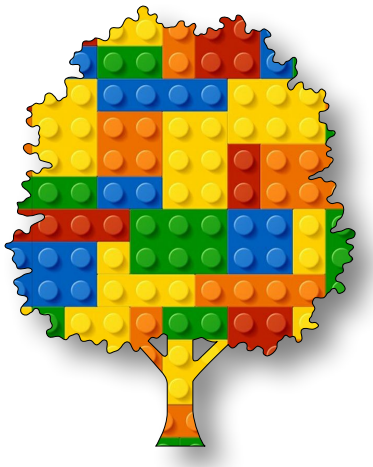


Table of Contents

Module G Unit #3 faceMesh

Using the faceMesh pre-trained model we can create some fun stuff with your face. We will be adding an image to your face so upload one to your sketch. In my case I have uploaded an image of some clouds from unsplash and superimposed it onto my face.

Sketch G3.1 index.html

Just a quick reminder to have the ml5.js library listed in the index.html file.

```
<!DOCTYPE html>
<html lang="en"><head>
  <script src="https://cdn.jsdelivr.net/npm/p5@2.2.3/lib/p5.js"></script>
  <script src="https://unpkg.com/ml5@1/dist/ml5.min.js"></script>
    <link rel="stylesheet" type="text/css" href="style.css">
    <meta charset="utf-8">
  </head>
  <body>
    <main>
    </main>
    <script src="sketch.js"></script>
</body></html>
```

Notes

Notice I am using p5.js version 2.2.3 I recommend that you use that version too if not already the default.

Sketch G3.2 our starting sketch

This is very similar to our hands sketch.

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

async function setup()
{
  createCanvas(640, 480, WEBGL)
  ml5.setBackend("webgl")
  faceMesh = await ml5.faceMesh({ flipped: true })
  video = createCapture(VIDEO, { flipped: true })
  video.hide()
  faceMesh.detectStart(video, gotFaces)
}

function gotFaces(results)
{
  faces = results
}
```

Notes

Nothing to see yet.

Sketch G3.3 points of view

We want to draw a pixel point() at each data or keypoint. The video will be commented out, if you want to see your face and the points on your face then just uncomment it.

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

async function setup()
{
  createCanvas(640, 480, WEBGL)
  ml5.setBackend("webgl")
  faceMesh = await ml5.faceMesh({ flipped: true })
  video = createCapture(VIDEO, { flipped: true })
  video.hide()
  faceMesh.detectStart(video, gotFaces)
}

function gotFaces(results)
{
  faces = results
}
```

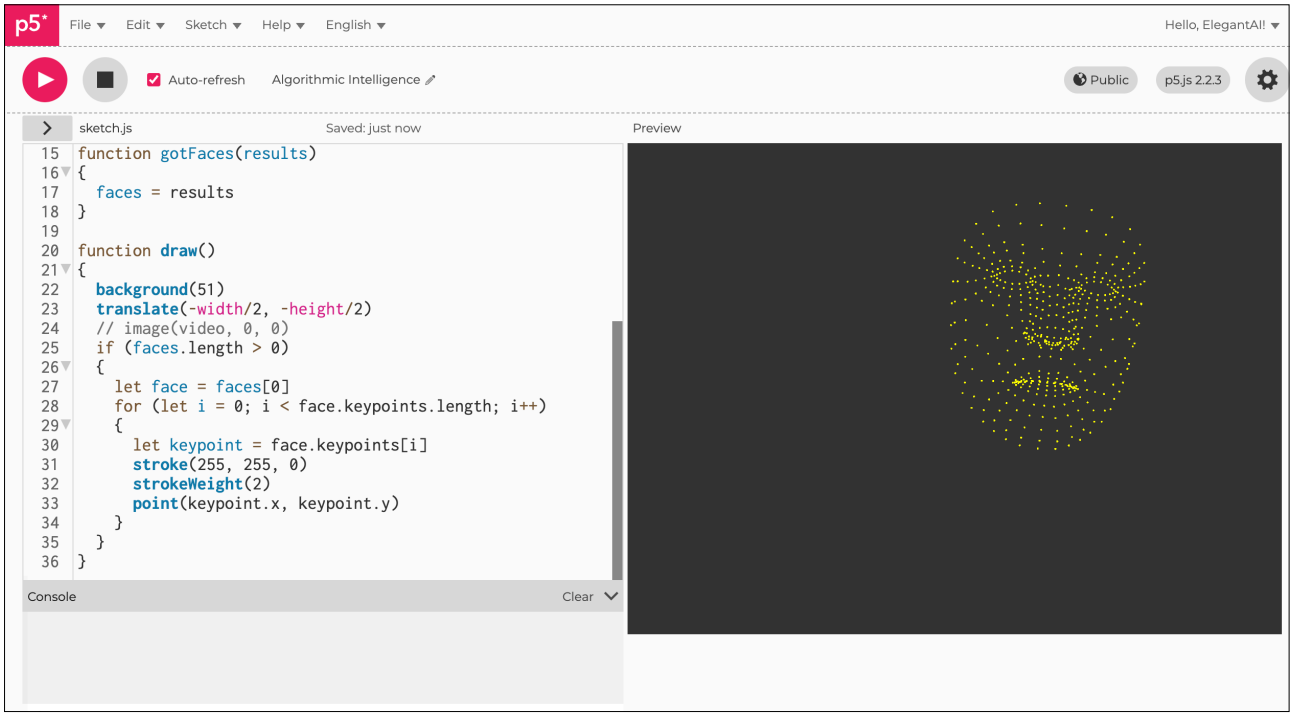
```
function draw()
{
  background(51)
  translate(-width/2, -height/2)
  // image(video, 0, 0)
  if (faces.length > 0)
  {
    let face = faces[0]
    for (let i = 0; i < face.keypoints.length; i++)
    {
      let keypoint = face.keypoints[i]
      stroke(255, 255, 0)
      strokeWeight(2)
      point(keypoint.x, keypoint.y)
    }
```

```
  }
}
```

Notes

I have run through this very quickly as it is so similar to the handPose models. There is plenty of interesting and fun stuff we can do with this model.

Figure G3.3



Sketch G3.4 step back

Remove the bulk of the code in draw. We want to draw the lips. So we need all the points that refer to the inner and outer lips in order. We add this as an array for each set.

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

async function setup()
{
  createCanvas(640, 480, WEBGL)
  ml5.setBackend("webgl")
  faceMesh = await ml5.faceMesh({ flipped: true })
  video = createCapture(VIDEO, { flipped: true })
  video.hide()
  faceMesh.detectStart(video, gotFaces)
}

function gotFaces(results)
{
  faces = results
}

function draw()
{
  background(51)
  translate(-width/2, -height/2)
  // image(video, 0, 0)
  if (faces.length > 0)
  {
    let face = faces[0]
  }
}
```

```
let lipsExterior = [267, 269, 270, 409, 291, 375, 321, 405, 314, 17, 84, 181, 91, 146, 61, 185, 40, 39, 37, 0]

let lipsInterior = [13, 312, 311, 310, 415, 308, 324, 318, 402, 317, 14, 87, 178, 88, 95, 78, 191, 80, 81, 82]
```

Notes

Somebody has very kindly created the lips arrays for us. I didn't do it. Whoever did that thank you.

Sketch G3.5 drawing outer lips

Using the outer lips array we cycle through them and draw the vertex at that data point.

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

async function setup()
{
  createCanvas(640, 480, WEBGL)
  ml5.setBackend("webgl")
  faceMesh = await ml5.faceMesh({ flipped: true })
  video = createCapture(VIDEO, { flipped: true })
  video.hide()
  faceMesh.detectStart(video, gotFaces)
}

function gotFaces(results)
{
  faces = results
}

function draw()
{
  background(51)
  translate(-width/2, -height/2)
  // image(video, 0, 0)
  if (faces.length > 0)
  {
    let face = faces[0]
    beginShape()
    for (let i = 0; i < lipsExterior.length; i++)
    {
      let index = lipsExterior[i]
      let keypoint = face.keypoints[index]
      stroke('red')
      strokeWeight(2)
```

```
      noFill()
      vertex(keypoint.x, keypoint.y)
    }
    endShape(CLOSE)
  }
}


let lipsExterior = [267, 269, 270, 409, 291, 375, 321, 405, 314, 17, 84, 181, 91, 146, 61, 185, 40, 39, 37, 0]
let lipsInterior = [13, 312, 311, 310, 415, 308, 324, 318, 402, 317, 14, 87, 178, 88, 95, 78, 191, 80, 81, 82]
```

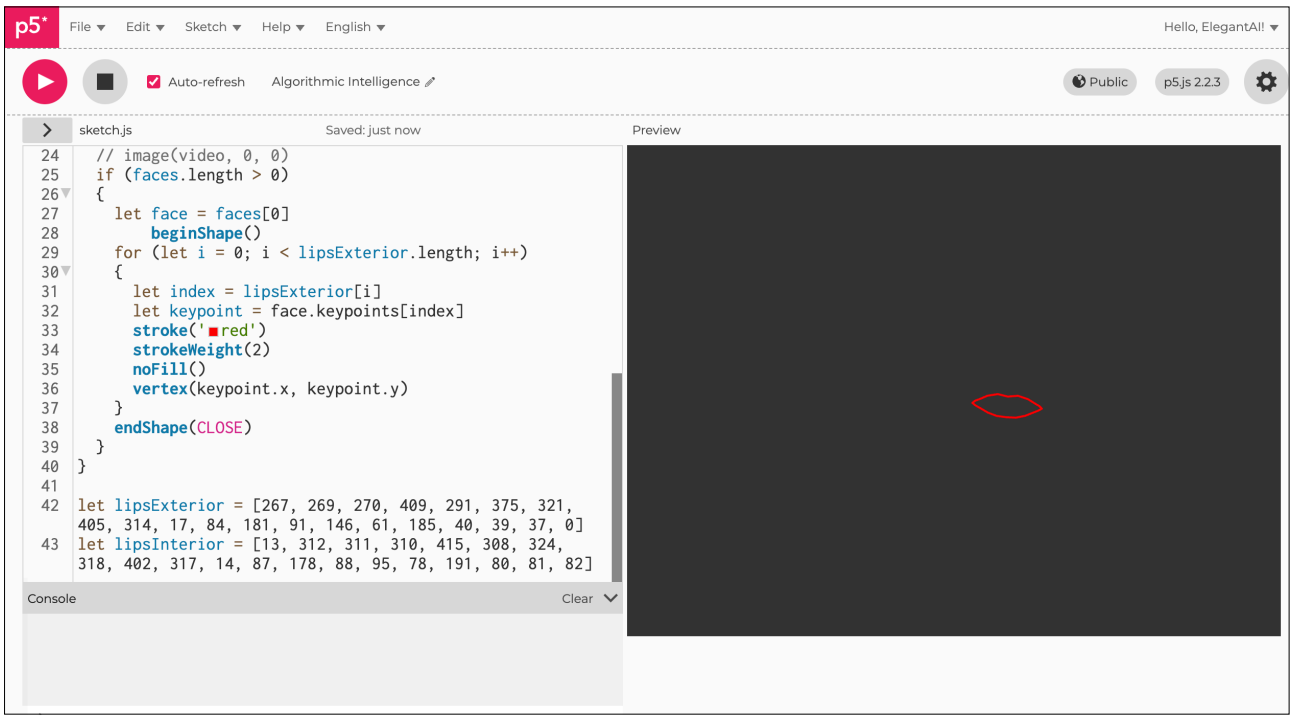
Notes

You should have your lips displayed and moving in sync.

Challenges

Change the colour
Add your face

Figure G3.5


p5*
File
Edit
Sketch
Help
English
Hello, ElegantAI!
Auto-refresh
Algorithmic Intelligence
Public
p5.js 2.2.3
sketch.js
Saved: just now
Preview
// image(video, 0, 0)
if (faces.length > 0)
{
let face = faces[0]
beginShape()
for (let i = 0; i < lipsExterior.length; i++)
{
let index = lipsExterior[i]
let keypoint = face.keypoints[index]
stroke('red')
strokeWeight(2)
noFill()
vertex(keypoint.x, keypoint.y)
}
endShape(CLOSE)
}
}
let lipsExterior = [267, 269, 270, 409, 291, 375, 321, 405, 314, 17, 84, 181, 91, 146, 61, 185, 40, 39, 37, 0]
let lipsInterior = [13, 312, 311, 310, 415, 308, 324, 318, 402, 317, 14, 87, 178, 88, 95, 78, 191, 80, 81, 82]
Console
Clear

Sketch G3.6 and the inner lips

The same again. This time cycle through the inner lips array drawing the vertex.

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

async function setup()
{
  createCanvas(640, 480, WEBGL)
  ml5.setBackend("webgl")
  faceMesh = await ml5.faceMesh({ flipped: true })
  video = createCapture(VIDEO, { flipped: true })
  video.hide()
  faceMesh.detectStart(video, gotFaces)
}

function gotFaces(results)
{
  faces = results
}

function draw()
{
  background(51)
  translate(-width/2, -height/2)
  // image(video, 0, 0)
  if (faces.length > 0)
  {
    let face = faces[0]
    beginShape()
    for (let i = 0; i < lipsExterior.length; i++)
    {
      let index = lipsExterior[i]
      let keypoint = face.keypoints[index]
      stroke('red')
      strokeWeight(2)
      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]
      stroke('white')
      strokeWeight(2)
      noFill()
      vertex(keypoint.x, keypoint.y)
    }
    endShape(CLOSE)
  }
}


let lipsExterior = [267, 269, 270, 409, 291, 375, 321, 405, 314, 17, 84, 181,
91, 146, 61, 185, 40, 39, 37, 0]
let lipsInterior = [13, 312, 311, 310, 415, 308, 324, 318, 402, 317, 14, 87,
178, 88, 95, 78, 191, 80, 81, 82]
```

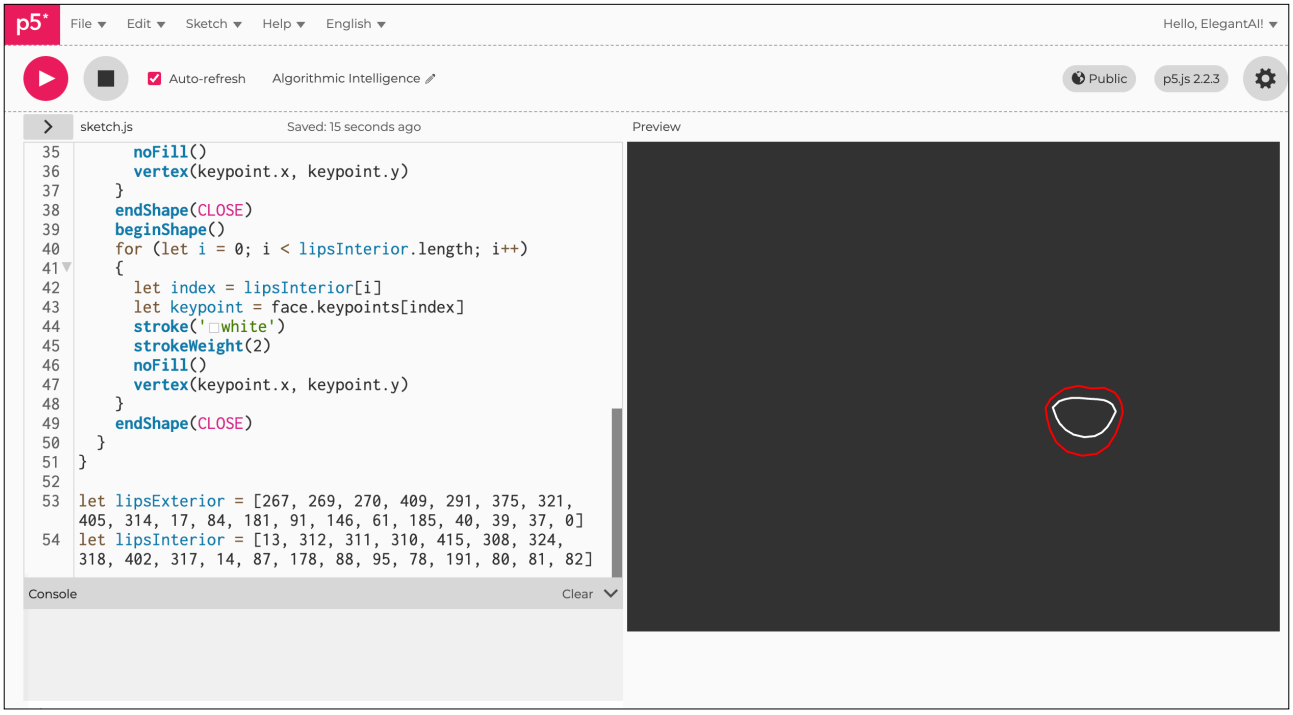
Notes

Should now have two sets of lips moving in sync.

Challenge

Can you think of a way to fill in between the inner and outer lips?

Figure G3.6


p5*
File
Edit
Sketch
Help
English
Hello, ElegantAI!
Auto-refresh
Algorithmic Intelligence
Public
p5.js 2.2.3
sketch.js
Saved: 15 seconds ago
Preview
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]
stroke('white')
strokeWeight(2)
noFill()
vertex(keypoint.x, keypoint.y)
}
endShape(CLOSE)
}
}
let lipsExterior = [267, 269, 270, 409, 291, 375, 321, 405, 314, 17, 84, 181, 91, 146, 61, 185, 40, 39, 37, 0]
let lipsInterior = [13, 312, 311, 310, 415, 308, 324, 318, 402, 317, 14, 87, 178, 88, 95, 78, 191, 80, 81, 82]
Console
Clear

Sketch G3.7 lip sync

We are going to add a nose (simple circle) that expands when you open and close your mouth. Why not! We need to work out when you are opening and closing your mouth.

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

async function setup()
{
  createCanvas(640, 480, WEBGL)
  ml5.setBackend("webgl")
  faceMesh = await ml5.faceMesh({ flipped: true })
  video = createCapture(VIDEO, { flipped: true })
  video.hide()
  faceMesh.detectStart(video, gotFaces)
}

function gotFaces(results)
{
  faces = results
}

function draw()
{
  background(51)
  translate(-width/2, -height/2)
  // image(video, 0, 0)
  if (faces.length > 0)
  {
    let face = faces[0]
    beginShape()
    for (let i = 0; i < lipsExterior.length; i++)
    {
      let index = lipsExterior[i]
      let keypoint = face.keypoints[index]
      stroke('red')
      strokeWeight(2)
```

```
      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]
      stroke('white')
      strokeWeight(2)
      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 lipsExterior = [267, 269, 270, 409, 291, 375, 321, 405, 314, 17, 84, 181,
91, 146, 61, 185, 40, 39, 37, 0]
let lipsInterior = [13, 312, 311, 310, 415, 308, 324, 318, 402, 317, 14, 87,
178, 88, 95, 78, 191, 80, 81, 82]
```

Notes

Working out the distance between point 13 and 14 for the interior lips.

Sketch G3.8 now the nose

We are going to add a new nose, nothing glamorous.

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

async function setup()
{
  createCanvas(640, 480, WEBGL)
  ml5.setBackend("webgl")
  faceMesh = await ml5.faceMesh({ flipped: true })
  video = createCapture(VIDEO, { flipped: true })
  video.hide()
  faceMesh.detectStart(video, gotFaces)
}

function gotFaces(results)
{
  faces = results
}

function draw()
{
  background(51)
  translate(-width/2, -height/2)
  // image(video, 0, 0)
  if (faces.length > 0)
  {
    let face = faces[0]
    beginShape()
    for (let i = 0; i < lipsExterior.length; i++)
    {
      let index = lipsExterior[i]
      let keypoint = face.keypoints[index]
      stroke('red')
      strokeWeight(2)
      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]
      stroke('white')
      strokeWeight(2)
      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)
    noStroke()
    let nose = face.keypoints[19]
    fill('yellow')
    circle(nose.x, nose.y, d)
  }
}


let lipsExterior = [267, 269, 270, 409, 291, 375, 321, 405, 314, 17, 84, 181,
91, 146, 61, 185, 40, 39, 37, 0]
let lipsInterior = [13, 312, 311, 310, 415, 308, 324, 318, 402, 317, 14, 87,
178, 88, 95, 78, 191, 80, 81, 82]
```

Notes

Looks even better with the image of you being displayed.

Challenge

Could be a clown nose

Blowing Bubbles

We are now going to blow bubbles through our mouth. Rather we are going to emit bubbles from our mouth when we open our mouth. This has many similarities to the emoji emitter in the previous unit. We will need an emitter class and a particle class.

Sketch G3.9 adding the emitter

! Here we remove the nose.

We want to have bubbles emit from your mouth as you open and close your mouth. Similar to the emoji emitter with handPose. So the emitter class is almost identical except that the particle now had a d value for the diameter.

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

async function setup()
{
  createCanvas(640, 480, WEBGL)
  ml5.setBackend("webgl")
  faceMesh = await ml5.faceMesh({ flipped: true })
  video = createCapture(VIDEO, { flipped: true })
  video.hide()
  faceMesh.detectStart(video, gotFaces)
}

function gotFaces(results)
{
  faces = results
}

function draw()
{
  background(51)
  translate(-width/2, -height/2)
  // image(video, 0, 0)
  if (faces.length > 0)
  {
    let face = faces[0]
    beginShape()
    for (let i = 0; i < lipsExterior.length; i++)
    {
      let index = lipsExterior[i]
      let keypoint = face.keypoints[index]
```

```
      stroke('red')
      strokeWeight(2)
      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]
      stroke('white')
      strokeWeight(2)
      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)
    // noStroke()
    // let nose = face.keypoints[19]
    // fill('yellow')
    // circle(nose.x, nose.y, d)
  }
}


let lipsExterior = [267, 269, 270, 409, 291, 375, 321, 405, 314, 17, 84, 181,
91, 146, 61, 185, 40, 39, 37, 0]
let lipsInterior = [13, 312, 311, 310, 415, 308, 324, 318, 402, 317, 14, 87,
178, 88, 95, 78, 191, 80, 81, 82]


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

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

  run()
  {
    for (let i = this.particles.length - 1; i >= 0; i--)
    {
      this.particles[i].run()
      if (this.particles[i].expired())
      {
        this.particles.splice(i, 1)
      }
    }
  }
}
```

Notes

The emitter references the particle.

Sketch G3.10 adding the particle

Again this is very similar to the previous unit so not going to go through all of it in detail. We will be adding gravity as a force shortly.

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

async function setup()
{
  createCanvas(640, 480, WEBGL)
  ml5.setBackend("webgl")
  faceMesh = await ml5.faceMesh({ flipped: true })
  video = createCapture(VIDEO, { flipped: true })
  video.hide()
  faceMesh.detectStart(video, gotFaces)
}

function gotFaces(results)
{
  faces = results
}

function draw()
{
  background(51)
  translate(-width/2, -height/2)
  // image(video, 0, 0)
  if (faces.length > 0)
  {
    let face = faces[0]
    beginShape()
    for (let i = 0; i < lipsExterior.length; i++)
    {
      let index = lipsExterior[i]
      let keypoint = face.keypoints[index]
      stroke('red')
      strokeWeight(2)
```

```
      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]
      stroke('white')
      strokeWeight(2)
      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 lipsExterior = [267, 269, 270, 409, 291, 375, 321, 405, 314, 17, 84, 181,
91, 146, 61, 185, 40, 39, 37, 0]
let lipsInterior = [13, 312, 311, 310, 415, 308, 324, 318, 402, 317, 14, 87,
178, 88, 95, 78, 191, 80, 81, 82]


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


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

```
  run()
  {
    for (let i = this.particles.length - 1; i >= 0; i--)
    {
      this.particles[i].run()
      if (this.particles[i].expired())
      {
        this.particles.splice(i, 1)
      }
    }
  }
}
```

```
class Particle
{
  constructor(x, y, d)
  {
    this.position = createVector(x, y)
    this.acceleration = createVector(0, 0)
    this.velocity = createVector(random(-2, 2), random(-2, 2))
    this.lifespan = 255
    this.r = d * 0.5
  }

  run()
  {
    this.move()
    this.show()
  }

  move()
  {
    this.velocity.add(this.acceleration)
    this.position.add(this.velocity)
    this.lifespan -= 2
    this.acceleration.mult(0)
  }
```

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

  expired()
  {
    return this.lifespan < 0
  }
}
```

Notes

The particle in this case is a simple circle. The fill is relative to half its lifespan.

Challenge

Use an emoji instead as the particle.

Sketch G3.11 now the bubbles

We need an emitter variable and create an emitter at some place initially then we calculate the half way point for the x and y values for each bubble as your mouth open, closes and moves. The bubble only emits if the mouth is open more than 20 pixels. To have it slow down a bit we use random(1) so that it emits just 75% of the time.

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

async function setup()
{
  createCanvas(640, 480, WEBGL)
  ml5.setBackend("webgl")
  faceMesh = await ml5.faceMesh({ flipped: true })
  video = createCapture(VIDEO, { flipped: true })
  video.hide()
  faceMesh.detectStart(video, gotFaces)
  emitter = new Emitter(300, 300)
}

function gotFaces(results)
{
  faces = results
}

function draw()
{
  background(51)
  translate(-width/2, -height/2)
  // image(video, 0, 0)
  if (faces.length > 0)
  {
    let face = faces[0]
    beginShape()
    for (let i = 0; i < lipsExterior.length; i++)
    {
```

```
      let index = lipsExterior[i]
      let keypoint = face.keypoints[index]
      stroke('red')
      strokeWeight(2)
      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]
      stroke('white')
      strokeWeight(2)
      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
    if (d > 20 && random(1) < 0.25)
    {
      emitter.addParticle(x, y, d)
    }
  }
  emitter.run()
}


let lipsExterior = [267, 269, 270, 409, 291, 375, 321, 405, 314, 17, 84, 181,
91, 146, 61, 185, 40, 39, 37, 0]
let lipsInterior = [13, 312, 311, 310, 415, 308, 324, 318, 402, 317, 14, 87,
178, 88, 95, 78, 191, 80, 81, 82]


class Emitter
{
```

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

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

  run()
  {
    for (let i = this.particles.length - 1; i >= 0; i--)
    {
      this.particles[i].run()
      if (this.particles[i].expired())
      {
        this.particles.splice(i, 1)
      }
    }
  }
}

class Particle
{
  constructor(x, y, d)
  {
    this.position = createVector(x, y)
    this.acceleration = createVector(0, 0)
    this.velocity = createVector(random(-2, 2), random(-2, 2))
    this.lifespan = 255
    this.r = d * 0.5
  }

  run()
  {
    this.move()
```

```
    this.show()
  }

  move()
  {
    this.velocity.add(this.acceleration)
    this.position.add(this.velocity)
    this.lifespan -= 2
    this.acceleration.mult(0)
  }

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

  expired()
  {
    return this.lifespan < 0
  }
}
```

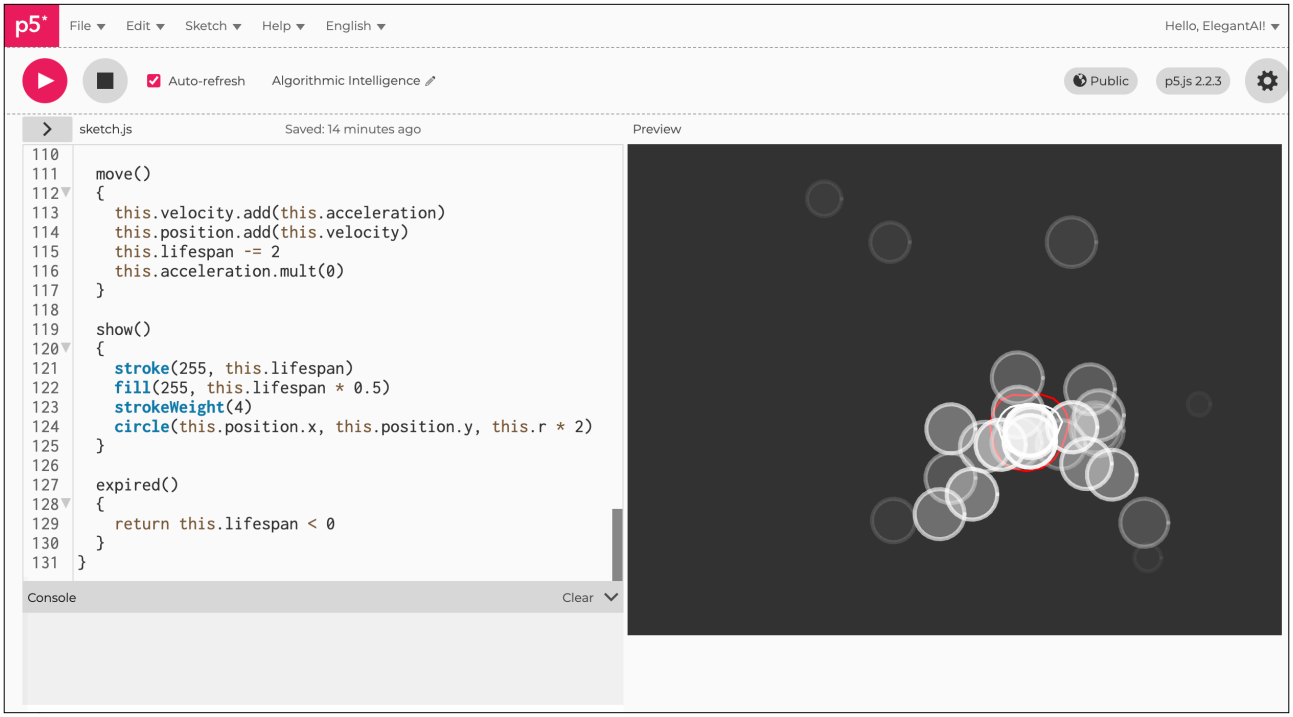
Notes

When the condition is met we add a particle to the emitter and we run the emitter. You should get bubbles coming out of your mouth. Notice that they go in all directions.

Challenge

You could do this without drawing the lips for a better effect.

Figure G3.11


p5*
File
Edit
Sketch
Help
English
Hello, ElegantAI!
Auto-refresh
Algorithmic Intelligence
Public
p5.js 2.2.3
sketch.js
Saved: 14 minutes ago
Preview
110
111 move()
112 {
113 this.velocity.add(this.acceleration)
114 this.position.add(this.velocity)
115 this.lifespan -= 2
116 this.acceleration.mult(0)
117 }
118
119 show()
120 {
121 stroke(255, this.lifespan)
122 fill(255, this.lifespan * 0.5)
123 strokeWeight(4)
124 circle(this.position.x, this.position.y, this.r * 2)
125 }
126
127 expired()
128 {
129 return this.lifespan < 0
130 }
131 }
Console
Clear

Sketch G3.12 negative gravity

It would be better if the bubbles rose upwards. So let us add some negative gravity!

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

async function setup()
{
  createCanvas(640, 480, WEBGL)
  ml5.setBackend("webgl")
  faceMesh = await ml5.faceMesh({ flipped: true })
  video = createCapture(VIDEO, { flipped: true })
  video.hide()
  faceMesh.detectStart(video, gotFaces)
  emitter = new Emitter(300, 300)
}

function gotFaces(results)
{
  faces = results
}

function draw()
{
  background(51)
  translate(-width/2, -height/2)
  // image(video, 0, 0)
  if (faces.length > 0)
  {
    let face = faces[0]
    beginShape()
    for (let i = 0; i < lipsExterior.length; i++)
    {
      let index = lipsExterior[i]
      let keypoint = face.keypoints[index]
      stroke('red')
```

```
      strokeWeight(2)
      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]
      stroke('white')
      strokeWeight(2)
      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
    if (d > 20 && random(1) < 0.25)
    {
      emitter.addParticle(x, y, d)
    }
  }
  emitter.run()
}


let lipsExterior = [267, 269, 270, 409, 291, 375, 321, 405, 314, 17, 84, 181,
91, 146, 61, 185, 40, 39, 37, 0]
let lipsInterior = [13, 312, 311, 310, 415, 308, 324, 318, 402, 317, 14, 87,
178, 88, 95, 78, 191, 80, 81, 82]


class Emitter
{
  constructor(x, y)
  {
    this.origin = createVector(x, y)
```

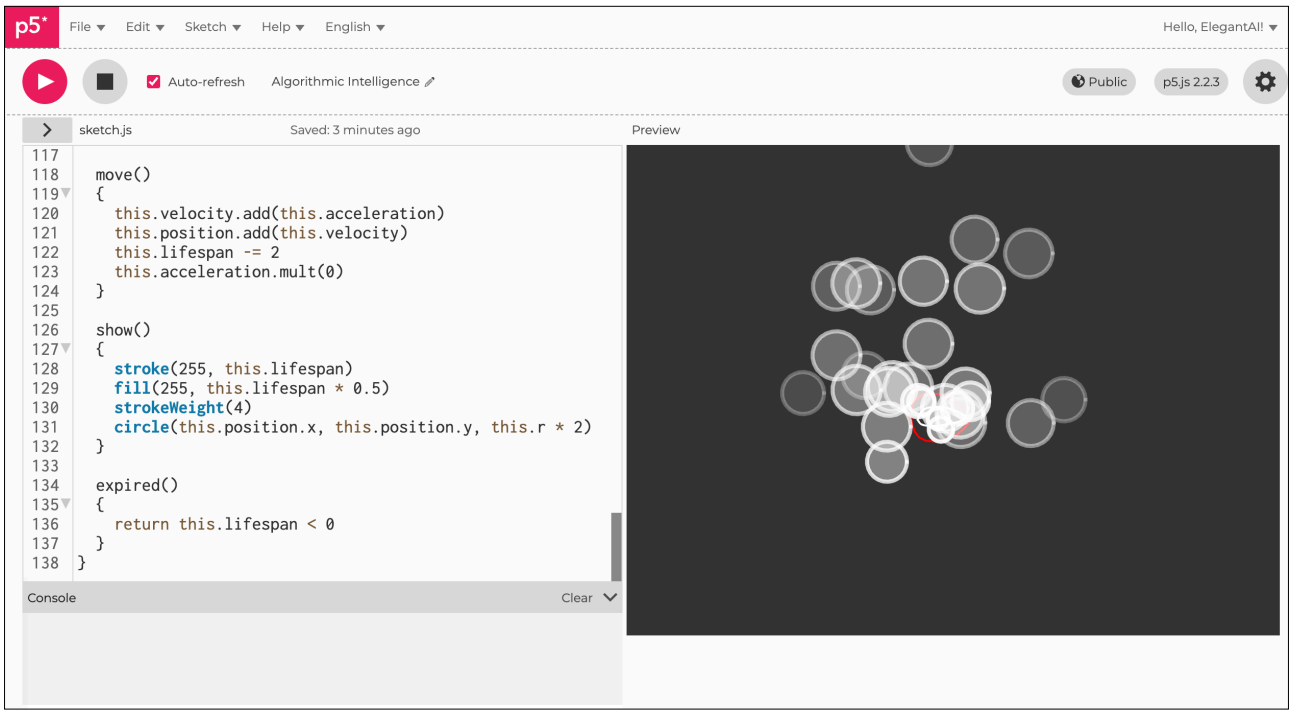
```
    this.particles = []
  }

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

  run()
  {
    for (let i = this.particles.length - 1; i >= 0; i--)
    {
      this.particles[i].run()
      if (this.particles[i].expired())
      {
        this.particles.splice(i, 1)
      }
    }
  }
}

class Particle
{
  constructor(x, y, d)
  {
    this.position = createVector(x, y)
    this.acceleration = createVector(0, 0)
    this.velocity = createVector(random(-2, 2), random(-2, 2))
    this.lifespan = 255
    this.r = d * 0.5
  }

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

```
  }

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

  move()
  {
    this.velocity.add(this.acceleration)
    this.position.add(this.velocity)
    this.lifespan -= 2
    this.acceleration.mult(0)
  }

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

  expired()
  {
    return this.lifespan < 0
  }
}
```

Notes

Looks really nice now.

Figure G3.12


p5*
File
Edit
Sketch
Help
English
Hello, ElegantAI!
Auto-refresh
Algorithmic Intelligence
Public
p5.js 2.2.3
sketch.js
Saved: 3 minutes ago
Preview
117
118 move()
119 {
120 this.velocity.add(this.acceleration)
121 this.position.add(this.velocity)
122 this.lifespan -= 2
123 this.acceleration.mult(0)
124 }
125
126 show()
127 {
128 stroke(255, this.lifespan)
129 fill(255, this.lifespan * 0.5)
130 strokeWeight(4)
131 circle(this.position.x, this.position.y, this.r * 2)
132 }
133
134 expired()
135 {
136 return this.lifespan < 0
137 }
138 }
Console
Clear

Sketch G3.13 back to basics

Remove everything except below.

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

async function setup()
{
  createCanvas(640, 480, WEBGL)
  ml5.setBackend("webgl")
  faceMesh = await ml5.faceMesh({ flipped: true })
  video = createCapture(VIDEO, { flipped: true })
  video.hide()
  faceMesh.detectStart(video, gotFaces)
}

function gotFaces(results)
{
  faces = results
}

function draw()
{
  background(51)
  translate(-width/2, -height/2)
  // image(video, 0, 0)
  if (faces.length > 0)
  {

  }
}
```

Notes

Just a blank canvas.

Sketch G3.14 triangles

One of the features of faceMesh is that it has a built-in function to get the triangles that join all the data points on the face. We create a variable called triangles to hold that data (console log to see it). There are three data points for each triangle. There are 852 triangles with a total of 2,556 data points.

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

async function setup()
{
  createCanvas(640, 480, WEBGL)
  ml5.setBackend("webgl")
  faceMesh = await ml5.faceMesh({ flipped: true })
  video = createCapture(VIDEO, { flipped: true })
  video.hide()
  faceMesh.detectStart(video, gotFaces)
  triangles = faceMesh.getTriangles()
}

function gotFaces(results)
{
  faces = results
}

function draw()
{
  background(51)
  translate(-width/2, -height/2)
  // image(video, 0, 0)
  if (faces.length > 0)
  {
    let face = faces[0]
  }
}
```

Notes

You might see it running slower in certain circumstances.

Sketch G3.15 showing the triangles

There is a lot happening here. The triangles holds the array of arrays (2556 data points) for each triangle. We cycle through each triangle pull out the data points (a, b, c). We use beginShape(TRIANGLES) to draw these triangles as vertex().

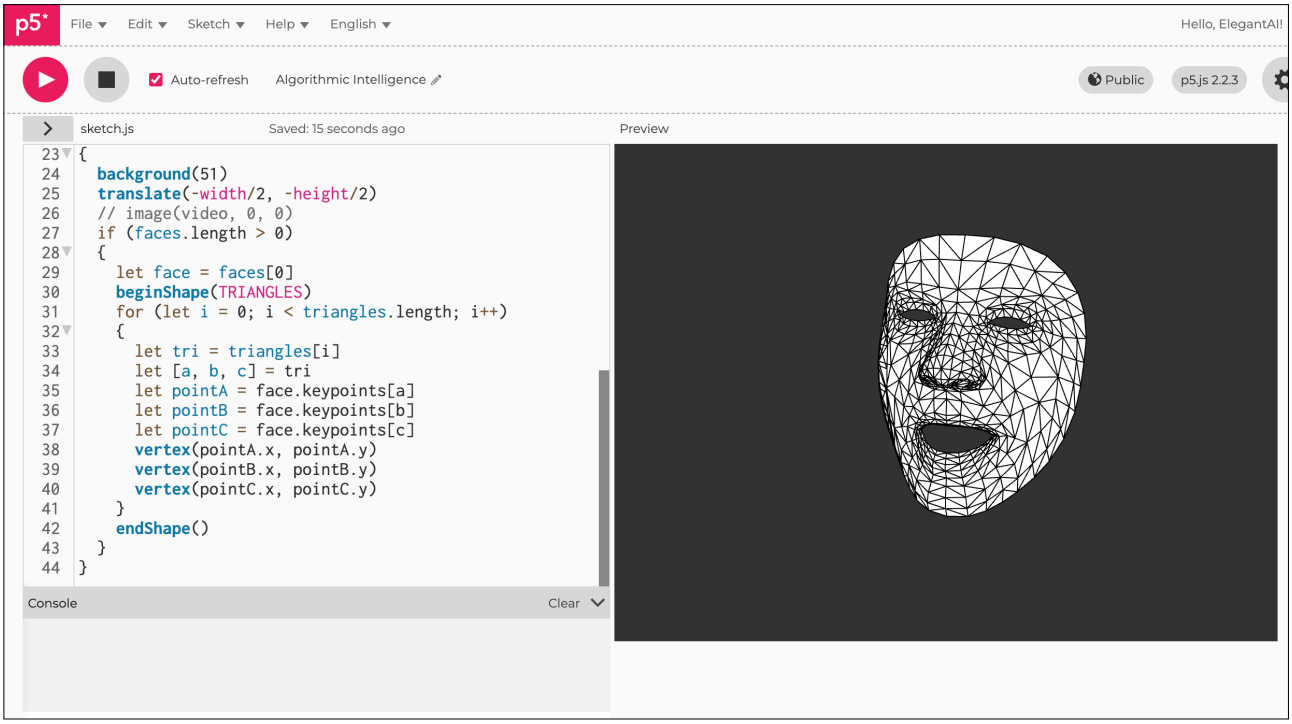
```
let video
let faceMesh
let faces = []
let triangles

async function setup()
{
  createCanvas(640, 480, WEBGL)
  ml5.setBackend("webgl")
  faceMesh = await ml5.faceMesh({ flipped: true })
  video = createCapture(VIDEO, { flipped: true })
  video.hide()
  faceMesh.detectStart(video, gotFaces)
  triangles = faceMesh.getTriangles()
}

function gotFaces(results)
{
  faces = results
}

function draw()
{
  background(51)
  translate(-width/2, -height/2)
  // image(video, 0, 0)
  if (faces.length > 0)
  {
    let face = faces[0]
    beginShape(TRIANGLES)
    for (let i = 0; i < triangles.length; i++)
    {
      let tri = triangles[i]
```

```
      let [a, b, c] = tri
      let pointA = face.keypoints[a]
      let pointB = face.keypoints[b]
      let pointC = face.keypoints[c]
      vertex(pointA.x, pointA.y)
      vertex(pointB.x, pointB.y)
      vertex(pointC.x, pointC.y)
    }
    endShape()
  }
}
```

Challenge

If you remove the triangle lines using noStroke()

Figure G3.15



Sketch G3.16 adding a mask

! Upload an image and rename it in the file and Sketch G3.accordingly. I have uploaded a cloud image as shown in the fig below.

We are going to use an image, any image will do, in my case I have downloaded an image of some clouds from unsplash but you can use any image or your own. The mask has co-ordinates x, y, z but these need to be mapped onto the image. The image is 2D and the co-ordinates are called u and v hence (u, v). We create a variable to hold the image (img) and the u, v co-ordinates (uvcoords). There is a function in faceMesh to get those co-ordinates, called getUVCoords().

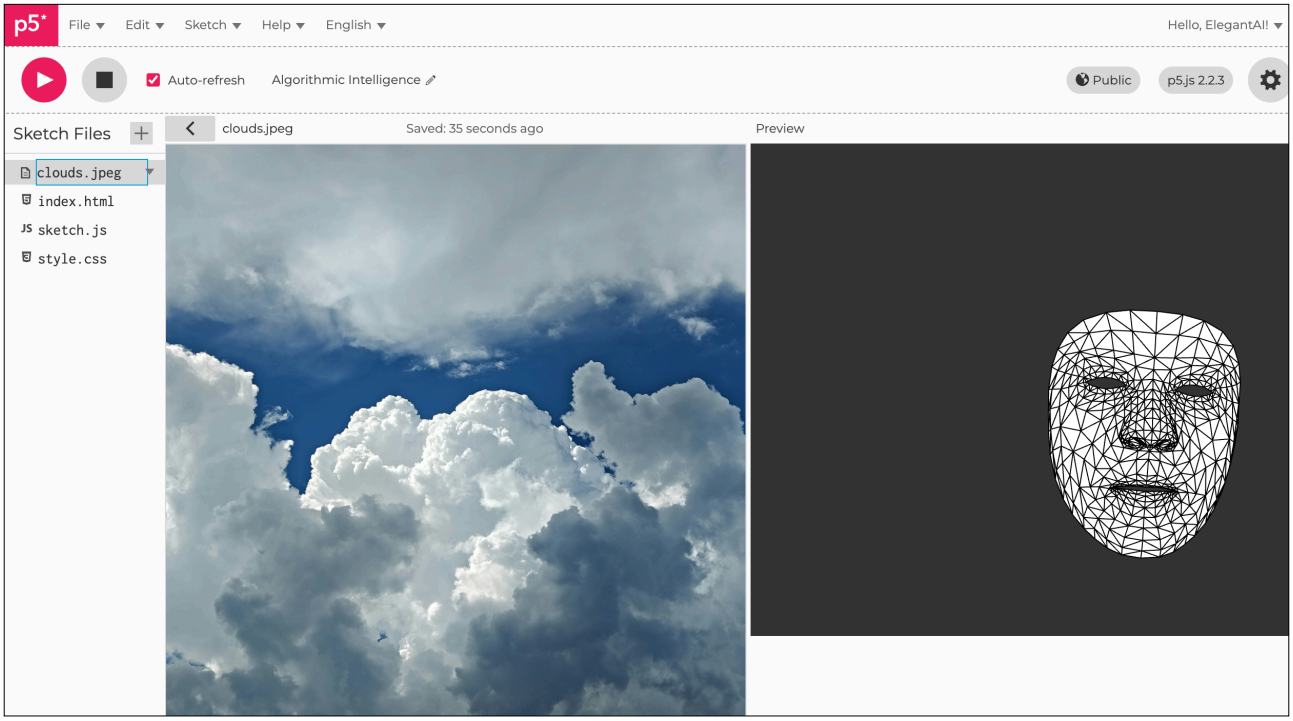
```
let video
let faceMesh
let faces = []
let triangles
let uvcoords
let img

async function setup()
{
  createCanvas(640, 480, WEBGL)
  ml5.setBackend("webgl")
  faceMesh = await ml5.faceMesh({ flipped: true })
  img = await loadImage("clouds.jpeg")
  video = createCapture(VIDEO, { flipped: true })
  video.hide()
  faceMesh.detectStart(video, gotFaces)
  triangles = faceMesh.getTriangles()
  uvcoords = faceMesh.getUVCoords()
}

function gotFaces(results)
{
  faces = results
}

function draw()
{
  background(51)
```

```
  translate(-width/2, -height/2)
  // image(video, 0, 0)
  if (faces.length > 0)
  {
    let face = faces[0]
    beginShape(TRIANGLES)
    for (let i = 0; i < triangles.length; i++)
    {
      let tri = triangles[i]
      let [a, b, c] = tri
      let pointA = face.keypoints[a]
      let pointB = face.keypoints[b]
      let pointC = face.keypoints[c]
      vertex(pointA.x, pointA.y)
      vertex(pointB.x, pointB.y)
      vertex(pointC.x, pointC.y)
    }
    endShape()
  }
}
```

Notes

This is just the beginning and is straightforward. You can use any image of anything, I have just used clouds for a effect.

Figure G3.16


p5*
File
Edit
Sketch
Help
English
Hello, ElegantAI!
Auto-refresh
Algorithmic Intelligence
Public
p5.js 2.2.3
Sketch Files
clouds.jpeg
index.html
sketch.js
style.css
clouds.jpeg
Saved: 35 seconds ago
Preview

Sketch G3.17 texture mapping

We now apply texture mapping to the detected face mesh.

```
let video
let faceMesh
let faces = []
let triangles
let uvcoords
let img

async function setup()
{
  createCanvas(640, 480, WEBGL)
  ml5.setBackend("webgl")
  faceMesh = await ml5.faceMesh({ flipped: true })
  img = await loadImage("clouds.jpeg")
  video = createCapture(VIDEO, { flipped: true })
  video.hide()
  faceMesh.detectStart(video, gotFaces)
  triangles = faceMesh.getTriangles()
  uvcoords = faceMesh.getUVCoords()
}

function gotFaces(results)
{
  faces = results
}

function draw()
{
  background(51)
  translate(-width/2, -height/2)
  // image(video, 0, 0)
  if (faces.length > 0)
  {
    let face = faces[0]
    texture(img)
    textureMode(NORMAL)
```

```
    noStroke()
    beginShape(TRIANGLES)
    for (let i = 0; i < triangles.length; i++)
    {
      let tri = triangles[i]
      let [a, b, c] = tri
      let pointA = face.keypoints[a]
      let pointB = face.keypoints[b]
      let pointC = face.keypoints[c]
      vertex(pointA.x, pointA.y)
      vertex(pointB.x, pointB.y)
      vertex(pointC.x, pointC.y)
    }
    endShape()
  }
}
```

Notes

You get a distorted image on the triangles which is because we haven't finished yet.

Sketch G3.18 mapping uv coordinates

Retrieve the corresponding UV coordinates for texture mapping and then define the triangle with both position (x, y) and UV texture coordinates.

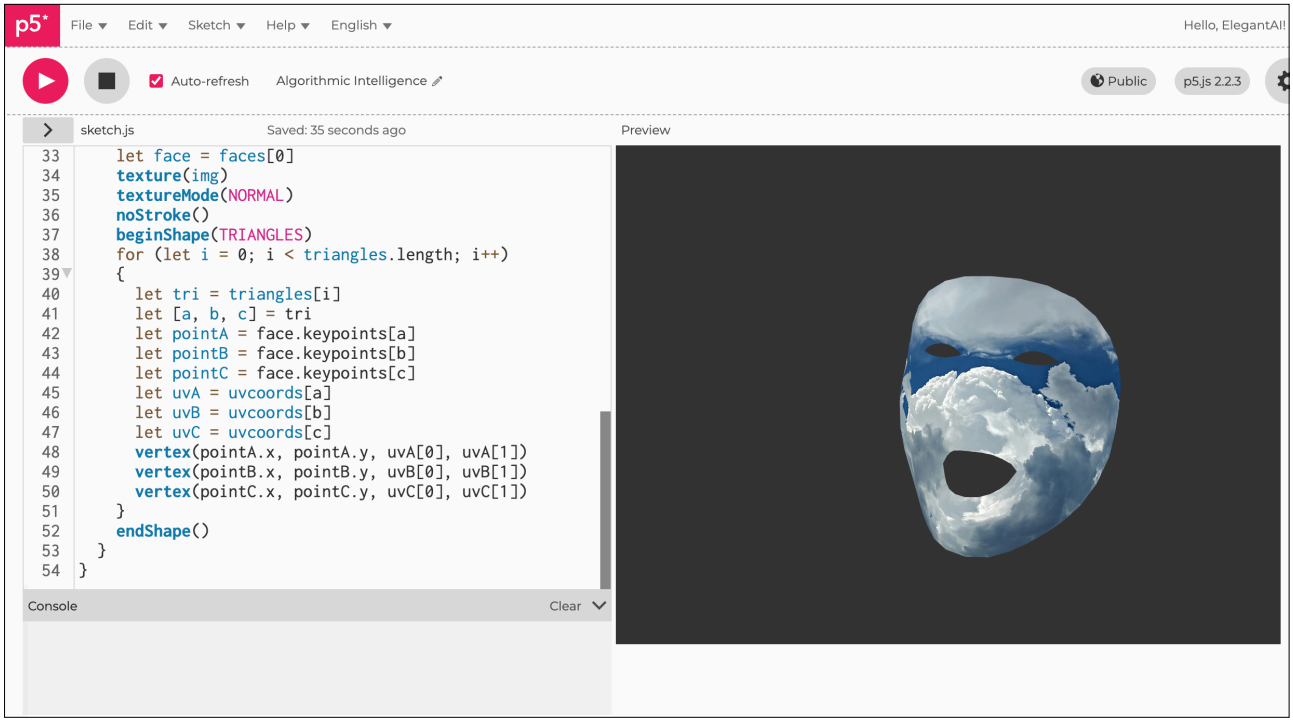
```
let video
let faceMesh
let faces = []
let triangles
let uvcoords
let img

async function setup()
{
  createCanvas(640, 480, WEBGL)
  ml5.setBackend("webgl")
  faceMesh = await ml5.faceMesh({ flipped: true })
  img = await loadImage("clouds.jpeg")
  video = createCapture(VIDEO, { flipped: true })
  video.hide()
  faceMesh.detectStart(video, gotFaces)
  triangles = faceMesh.getTriangles()
  uvcoords = faceMesh.getUVCoords()
}

function gotFaces(results)
{
  faces = results
}

function draw()
{
  background(51)
  translate(-width/2, -height/2)
  // image(video, 0, 0)
  if (faces.length > 0)
  {
    let face = faces[0]
    texture(img)
```

```
    textureMode(NORMAL)
    noStroke()
    beginShape(TRIANGLES)
    for (let i = 0; i < triangles.length; i++)
    {
      let tri = triangles[i]
      let [a, b, c] = tri
      let pointA = face.keypoints[a]
      let pointB = face.keypoints[b]
      let pointC = face.keypoints[c]
      let uvA = uvcoords[a]
      let uvB = uvcoords[b]
      let uvC = uvcoords[c]
      vertex(pointA.x, pointA.y, uvA[0], uvA[1])
      vertex(pointB.x, pointB.y, uvB[0], uvB[1])
      vertex(pointC.x, pointC.y, uvC[0], uvC[1])
    }
    endShape()
  }
}
```

Notes

We have a cloud mask.

Figure G3.18


p5*
File
Edit
Sketch
Help
English
Hello, ElegantAI!
Auto-refresh
Algorithmic Intelligence
Public
p5.js 2.2.3
sketch.js
Saved: 35 seconds ago
Preview
let face = faces[0]
texture(img)
textureMode(NORMAL)
noStroke()
beginShape(TRIANGLES)
for (let i = 0; i < triangles.length; i++)
{
let tri = triangles[i]
let [a, b, c] = tri
let pointA = face.keypoints[a]
let pointB = face.keypoints[b]
let pointC = face.keypoints[c]
let uvA = uvcoords[a]
let uvB = uvcoords[b]
let uvC = uvcoords[c]
vertex(pointA.x, pointA.y, uvA[0], uvA[1])
vertex(pointB.x, pointB.y, uvB[0], uvB[1])
vertex(pointC.x, pointC.y, uvC[0], uvC[1])
}
endShape()
}
}
Console
Clear