

*LABORATÓRIO DE SOM E IMAGEM*  
*2016/2017*

*INTRODUÇÃO À PROGRAMAÇÃO  
COM PROCESSING*

*RODRIGO CARVALHO  
WWW.VISIOPHONE-LAB.COM*

Pattern | Processing 3.0.1

Pattern

```
background(200);
}

void draw() {
    // Call the variableEllipse() method and send it the
    // parameters for the current mouse position
    // and the previous mouse position
    variableEllipse(mouseX, mouseY, pmouseX, pmouseY);
}

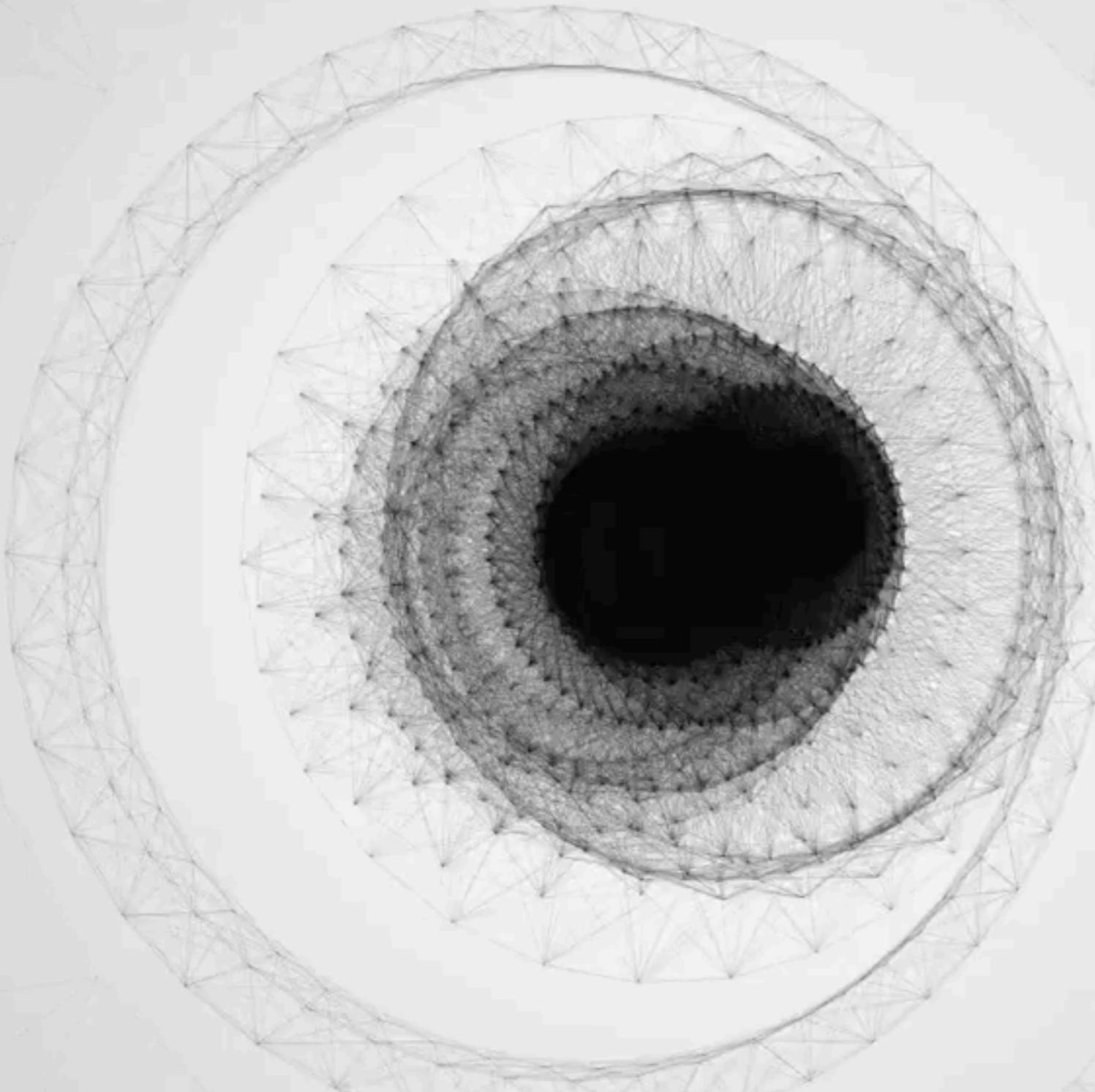
// The simple method variableEllipse() was created specifically
// for this program. It calculates the speed of the mouse
// and draws a small ellipse if the mouse is moving slowly
// and draws a large ellipse if the mouse is moving quickly

void variableEllipse(int x, int y, int px, int py) {
    float speed = abs(x-px) + abs(y-py);
    stroke(150);
    fill(10);
    ellipse(x, y, speed*2, speed*2);
}
```

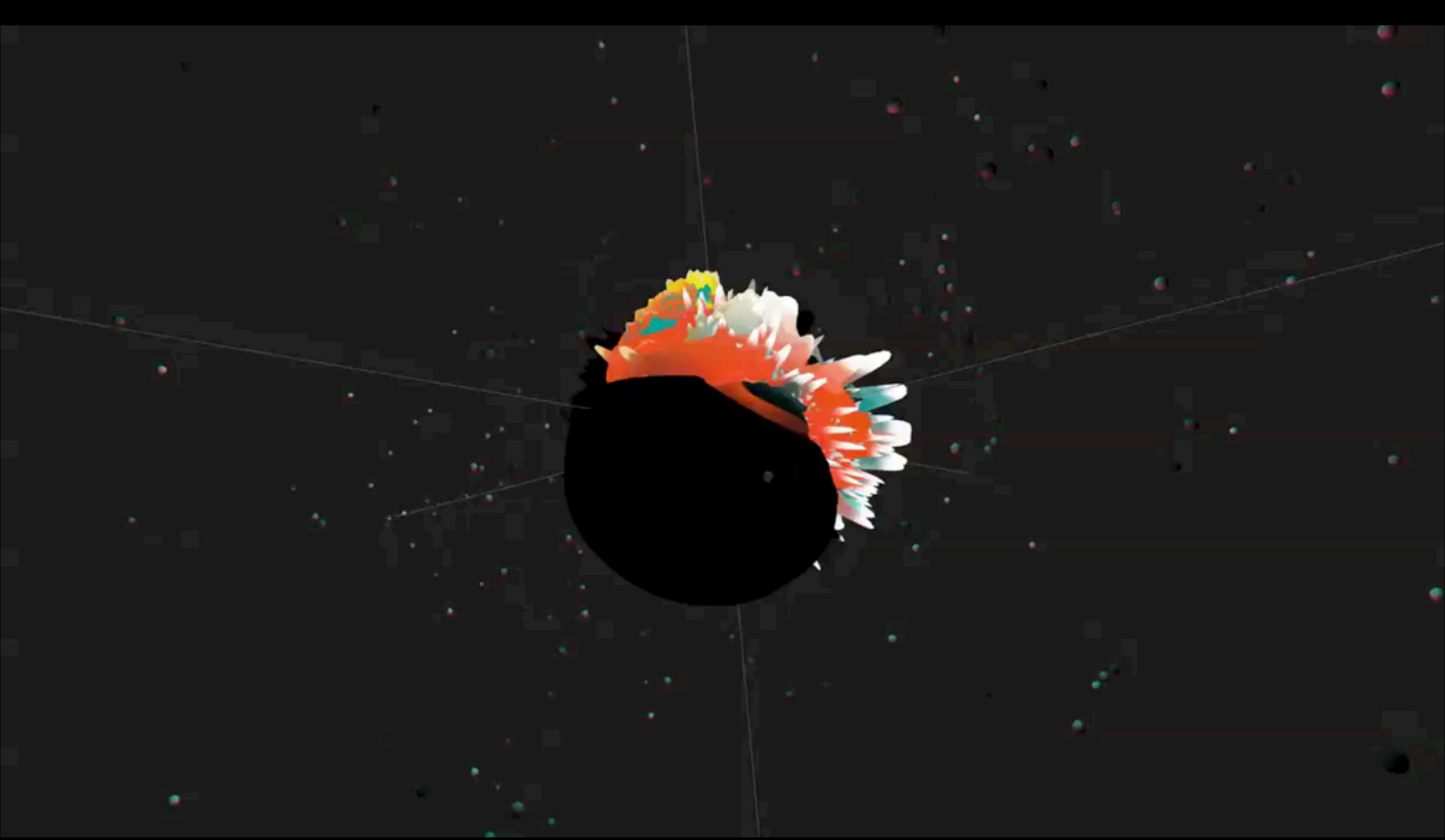
Pattern

Console Errors Updates 2

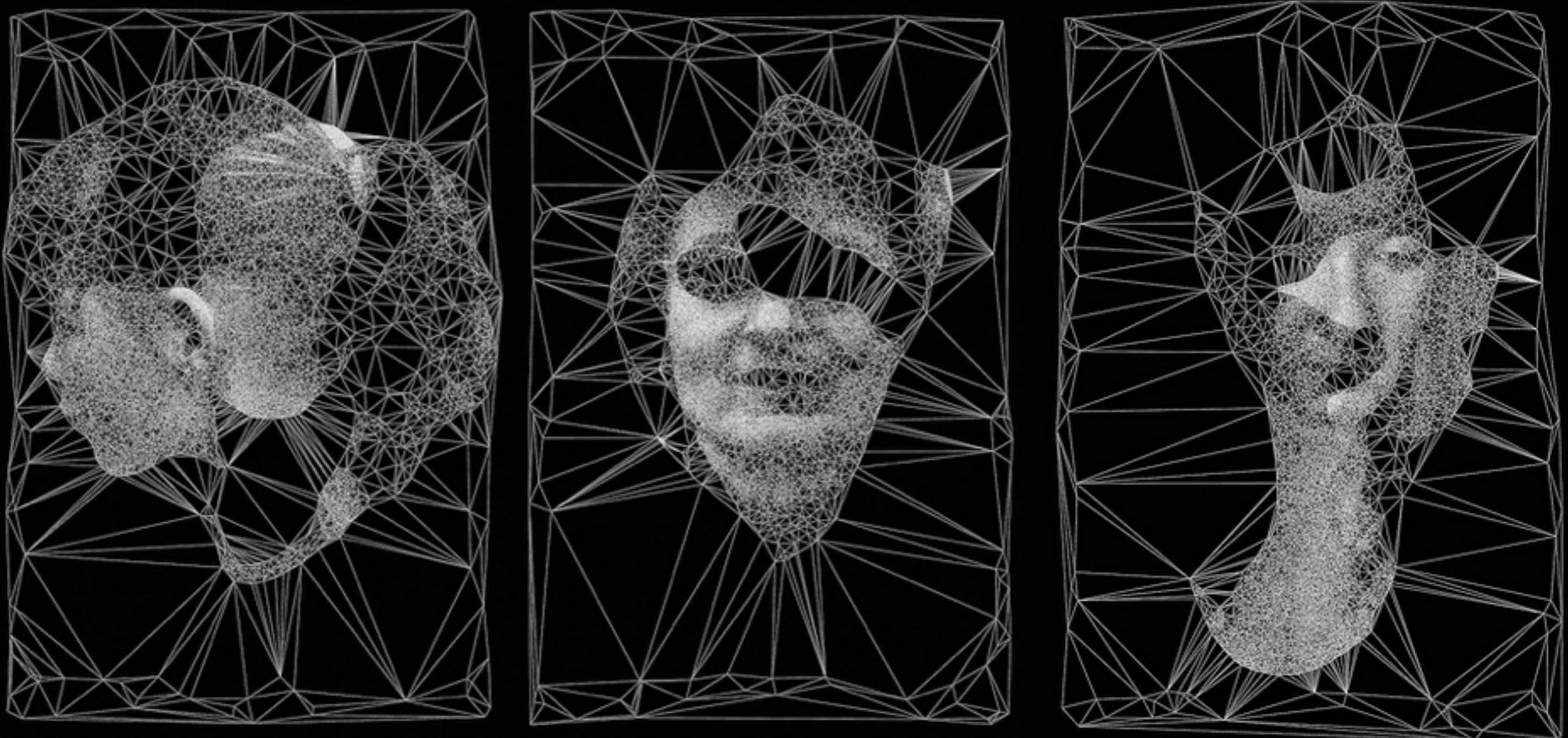
**PROCESSING | CREATIVE CODING | SOFTWARE ART**  
**DATA VISUALIZATION | LIVE VISUALS | GENERATIVE |**  
**INTERACTIVE INSTALLATIONS | AUDIOVISUAL ARTWORK |**  
**RANDOMNESS | . . .**



*AUDIO GENERATED OBJECTS*  
DIANA LANGE/ 2010



*THE DEEPEST OF SPACE*  
JOSHUA DAVIDS / 2015



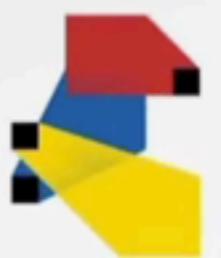
*GENERATIVE PORTRAITS*  
DIANA LANGE / 2013



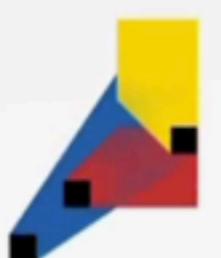
GENERATIVE PORTRAITS  
DIANA LANGE / 2013



*DEBUG - ART MADE BY INSECTS*  
EDHV / 2010



MIT MEDIA LAB



MIT MEDIA LAB



MIT MEDIA LAB



MIT MEDIA LAB



MIT MEDIA LAB



MIT MEDIA LAB



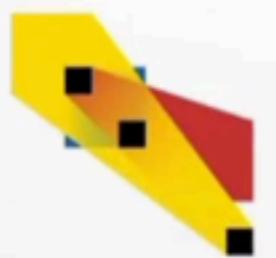
MIT MEDIA LAB



MIT MEDIA LAB



MIT MEDIA LAB



MIT MEDIA LAB

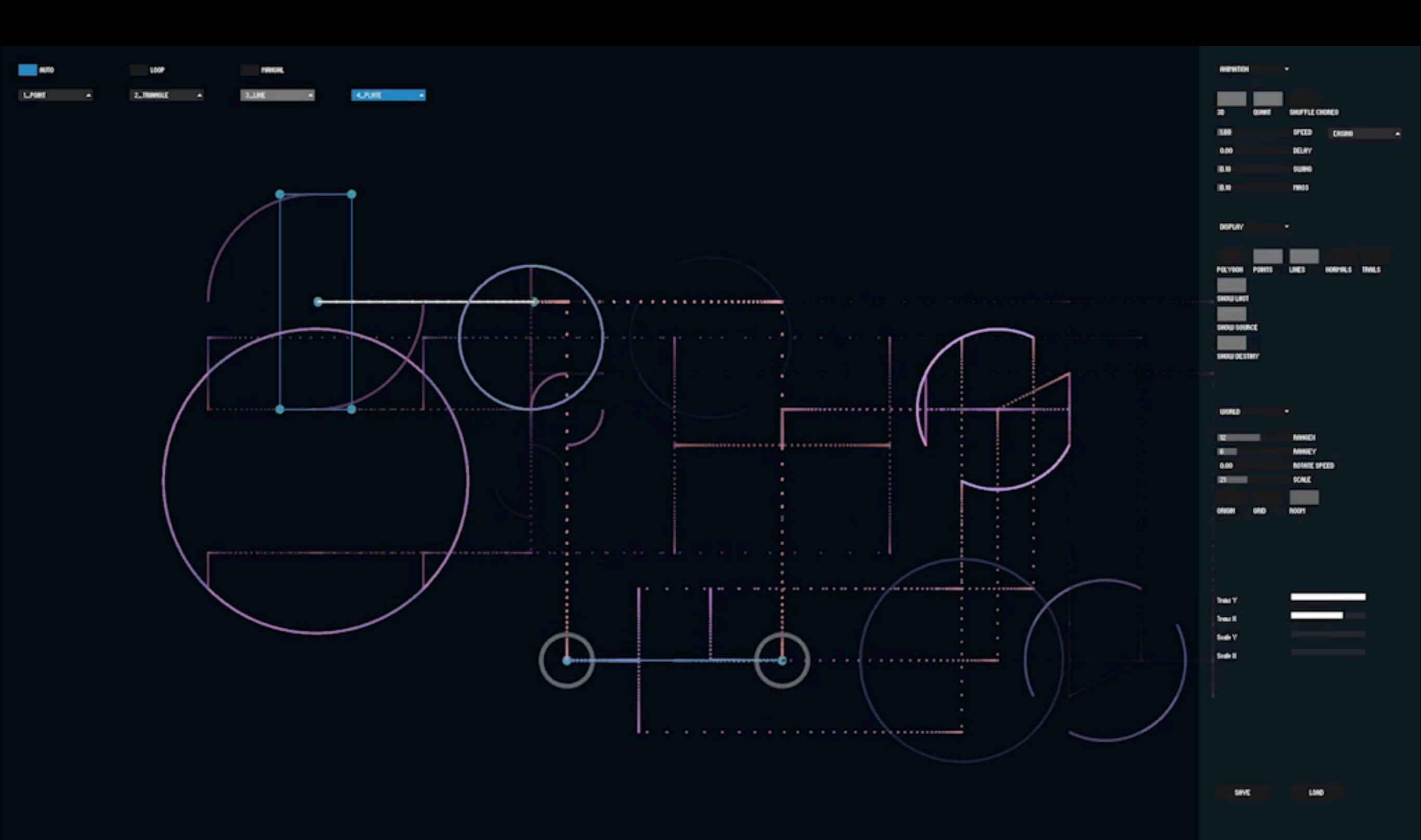


MIT MEDIA LAB



MIT MEDIA LAB

*MIT MEDIA LAB  
THEGREENEYL / 2011*



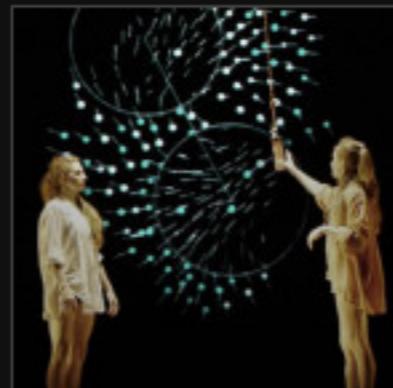
PATHFINDER  
WALTZ BINNAIRE + ONFORMATIVE / 2014



VAVA METROPLEX



KOBAYASHI



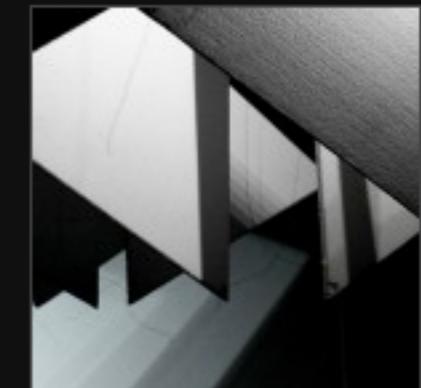
WITH OUI [2015]



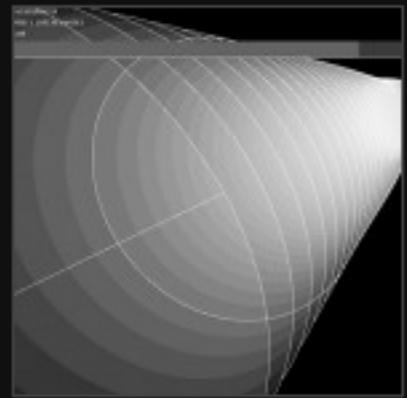
SCION/NY



AD MORTUOS



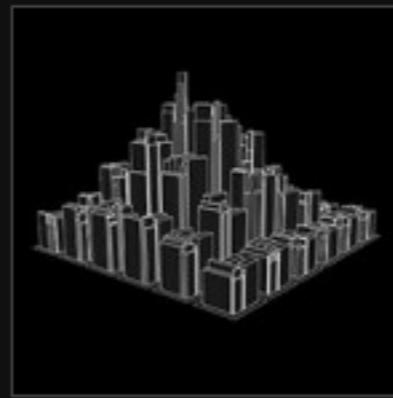
BC504+ALMAD' ARAME  
EXPLORATIVE  
RESIDENCY



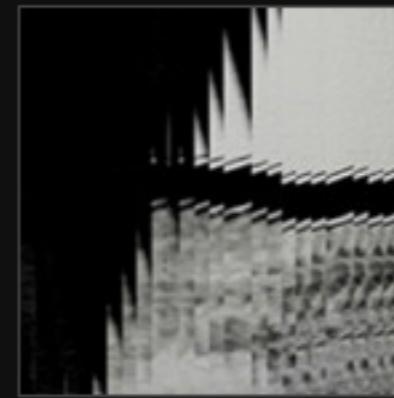
THE INTERPRETER



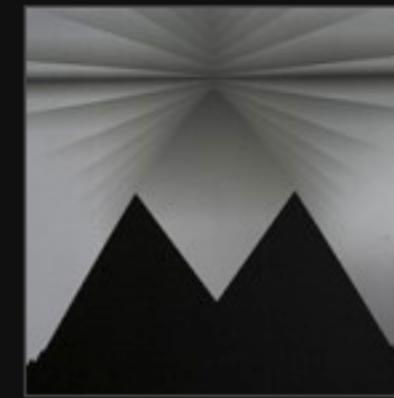
FLOATING  
SATELLITES



RE-ENVISION EP



DAMAGED DUCHESS



LINEAR SUBSPACE



SUBMERSIBLE  
COMPUTER CENTER



RE-ENVISION



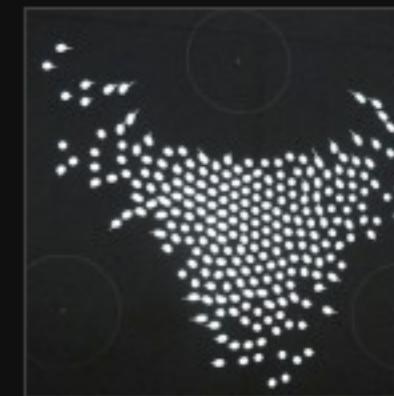
WEARABLE DRESS  
PROTOTYPE



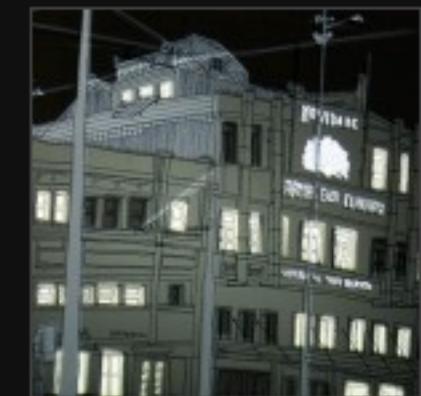
BREAKDOWN



VAVA



X GRAVITY



LUSCO FUSCO NA  
PRAÇA

*RODRIGO CARVALHO / [WWW.VISIOPHONE-LAB.COM](http://www.visiophone-lab.com)*



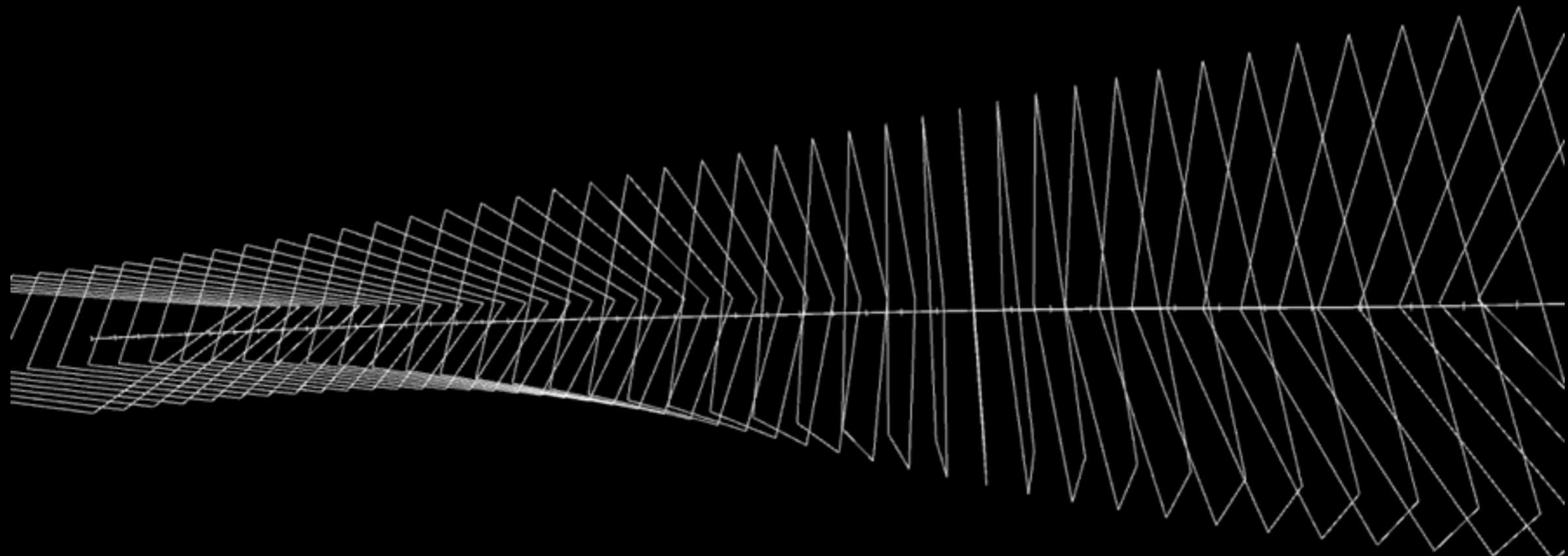
*REDUCTION/REFLECTION*

VISUALS: RODRIGO CARVALHO | MUSIC: @C / 2016



*THE INTERPRETER*  
RODRIGO C. + YAGO DE QUAY + SUN CHEN / 2014

THE INTERPRETER  
SPEED: 1 | FPS: 60 | MODE:3  
3897



*THE INTERPRETER*  
RODRIGO C.+ YAGO DE QUAY + SUN CHEN / 2014



*WITH OUI*  
EYES+EARS+FEET UT.AUSTIN / 2014



*WWW.PROCESSING.ORG*

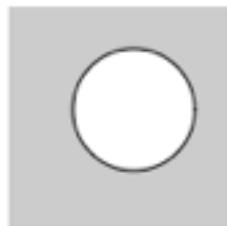
The image shows the Processing interface with the following components:

- PLAY/STOP**: A button at the top left of the code editor.
- Code Editor**: Displays the following sketch code:

```
1 // setup() runs first one time.  
2 void setup() {  
3  
4     // screen size width,height  
5     size(480, 820);  
6 }  
7  
8 // draw() loops continuously until you close  
9 void draw() {  
10  
11    // If mousePressed color= Black. If not color=White  
12    if (mousePressed) {  
13        fill(0);  
14    } else {  
15        fill(255);  
16    }  
17  
18    // ellipse, poX, posY, size, size)  
19    ellipse(mouseX, mouseY, 80, 80);  
20 }  
21 }  
22  
23
```
- Canvas**: Shows a 3D-like wireframe sculpture composed of many nested ellipses.
- NOTAS**: An annotation pointing to the condition in line 12: `// If mousePressed color= Black. If not color=White`.
- COR**: An annotation pointing to the fill color assignment in line 15: `fill(255);`.
- FUNÇÃO GRÁFICA: ELLIPSE**: An annotation pointing to the `ellipse` function call in line 19: `ellipse(mouseX, mouseY, 80, 80);`.
- CONSOLA MENSAGENS**: An annotation pointing to the message "Done saving." in the status bar.
- Console**: A tab at the bottom left.
- Errors**: A tab at the bottom left.
- Updates**: A tab at the bottom right with a notification count of 2.

PROCESSING INTERFACE

# ellipse()



```
ellipse(56, 46, 55, 55);
```

Draws an ellipse (oval) to the screen. An ellipse with equal width and height is a circle. By default, the first two parameters set the location, and the third and fourth parameters set the size. The origin may be changed with the `ellipseMode()` function.

`ellipse(a, b, c, d)`

**a** float: x-coordinate of the ellipse

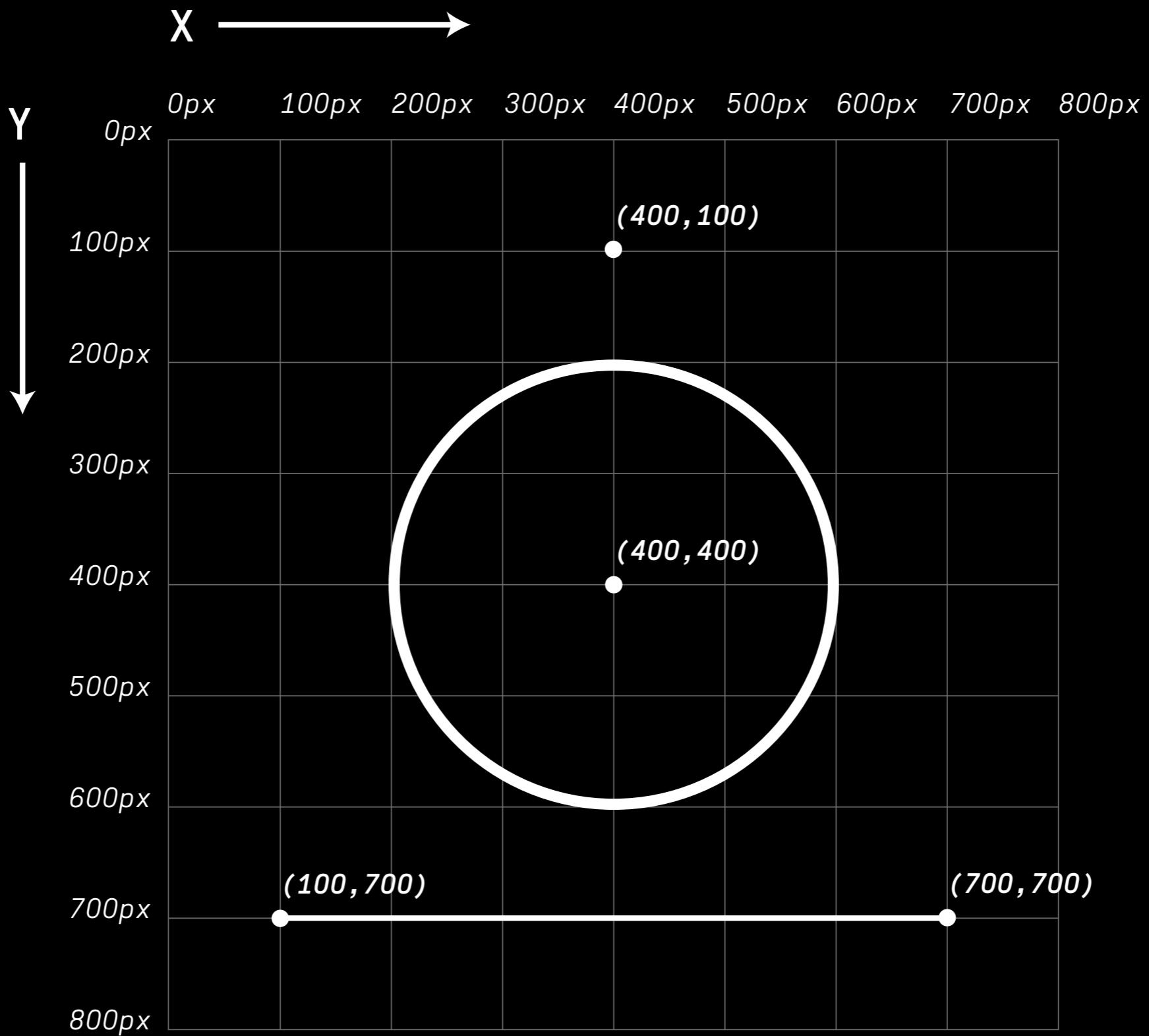
**b** float: y-coordinate of the ellipse

**c** float: width of the ellipse by default

**d** float: height of the ellipse by default

**Reference.** Processing was designed to be a flexible software sketchbook.

Structure	Shape	Color
() (parentheses)	createShape()	
, (comma)	loadShape()	Setting
. (dot)	PShape	background()
/* */ (multiline comment)		clear()
/**/ (doc comment)		colorMode()
// (comment)	2D Primitives	fill()
;(semicolon)	arc()	noFill()
= (assign)	ellipse()	noStroke()
[] (array access)	line()	stroke()
{ } (curly braces)	point()	
catch	quad()	Creating & Reading
class	rect()	alpha()
draw()	triangle()	blue()
exit()		brightness()
extends	Curves	color()
false	bezier()	green()
final	bezierDetail()	hue()
implements	bezierPoint()	lerpColor()
import	bezierTangent()	red()
loop()	curve()	saturation()
new	curveDetail()	
noLoop()	curvePoint()	Image
null	curveTangent()	createImage()
popStyle()	curveTightness()	PIImage
private		
	3D Primitives	



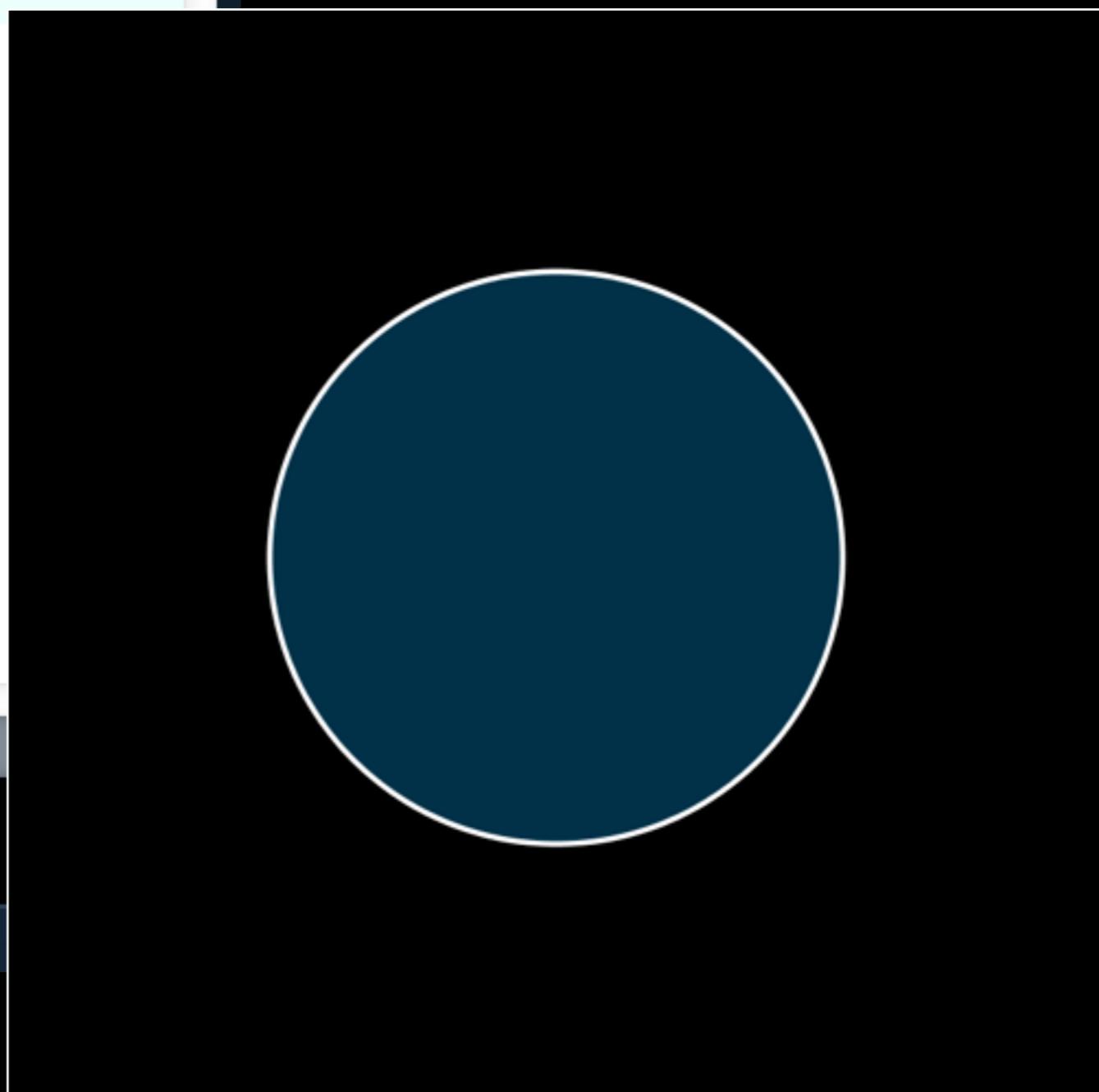
COORDENADAS  $(XX, YY)$

LSI\_005\_circuloSize ▾

```
1 // reads on the beginnig
2 void setup() {
3     //screen size
4     size(800, 800);
5 }
6
7 // loops / read forever
8 void draw() {
9
10 //background color
11 background(0,0,0);
12
13 //console print
14 println(mouseX);
15
16 //color
17 fill(0,50,70);
18 stroke(255,255,255);
19 strokeWeight(4);
20
21 //circle
22 ellipse (400,400,mouseX,mouseX);
23 }
24
25
```

51  
51  
51

Java ▾



Console Errors



Stop



Java ▾

## LSI\_011\_linhasMove\_fade ▾

```
1 //reads once. on start
2 void setup() {
3     // display size
4     size(800, 800);
5 }
6
7 //loops forever
8 void draw () {
9
10    //rect with the same size as the display
11    // with alpha () , for the fade effect
12    fill(0, 10);
13    noStroke();
14    rect(0, 0, width, height);
15
16    // Stroke Color. Changing with the Mouse movement
17    stroke(255, mouseX, mouseY);
18    // horizontal line. following the mouse
19    line(0+mouseX, mouseY, 800-mouseX, mouseY);
20
21    //printing mouseX + SPACE + mouseY
22    println(mouseX+" "+mouseY);
23 }
```

```
10 15
10 15
10 15
```

Console

Errors



```
// SCREEN SIZE //CONSOLE PRINT
size (width,height); println("hello");
ellipse(posX, posY, size,size); //DISPLAY DIMENSIONS
// CIRCLE / ELLIPSE //MOUSE POSITION
width;
height;
// COLOR //BACKGROUND COLOR
fill(R,G,B); background(R,G,B);
fill(R,G,B,A); mouseX;
fill(GREY); mouseY;
noFill();
// COLOR BORDER
// RECTANGLE
stroke(R,G,B); rectMode(CORNER);
stroke(R,G,B); rect(cornerX, cornerY, width, height);
stroke(R,G,B,A); rectMode(CENTER);
stroke(GREY); rect(centerX, centerY, width, height);
noStroke();
strokeWeight(nr);
```

*LINKS:*

// Official Processing website  
WWW.PROCESSING.ORG

// Archive Processing sketches. Sharing Code ;)  
WWW.OPENPROCESSING.ORG

//Blog/Portal Interactive Art  
WWW.CREATIVEAPPLICATIONS.NET

//Portfolio Rodrigo Carvalho  
WWW.VISIOPHONE-LAB.COM