15/02/2017

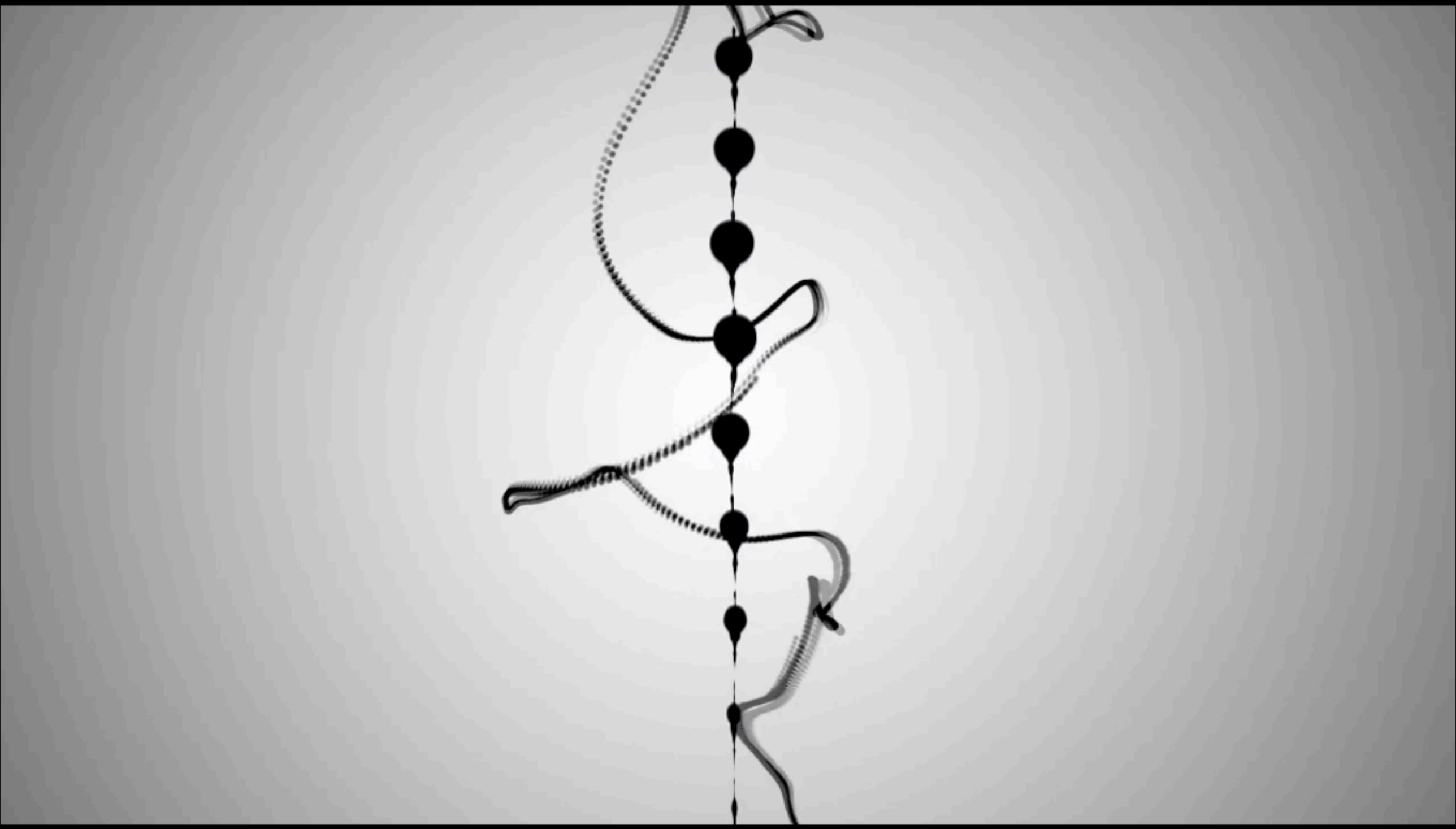
LABORATÓRIO DE SOM E IMAGEM
2016/2017

INTRODUÇÃO À PROGRAMAÇÃO COM PROCESSING

RODRIGO CARVALHO

MATERIAIS AULAS:

/GITHUB.COM/VISIOPHONE/LSI



SUPERCOLLIDER SHAPE

JOÃO MARTINHO MOURA/ 2011

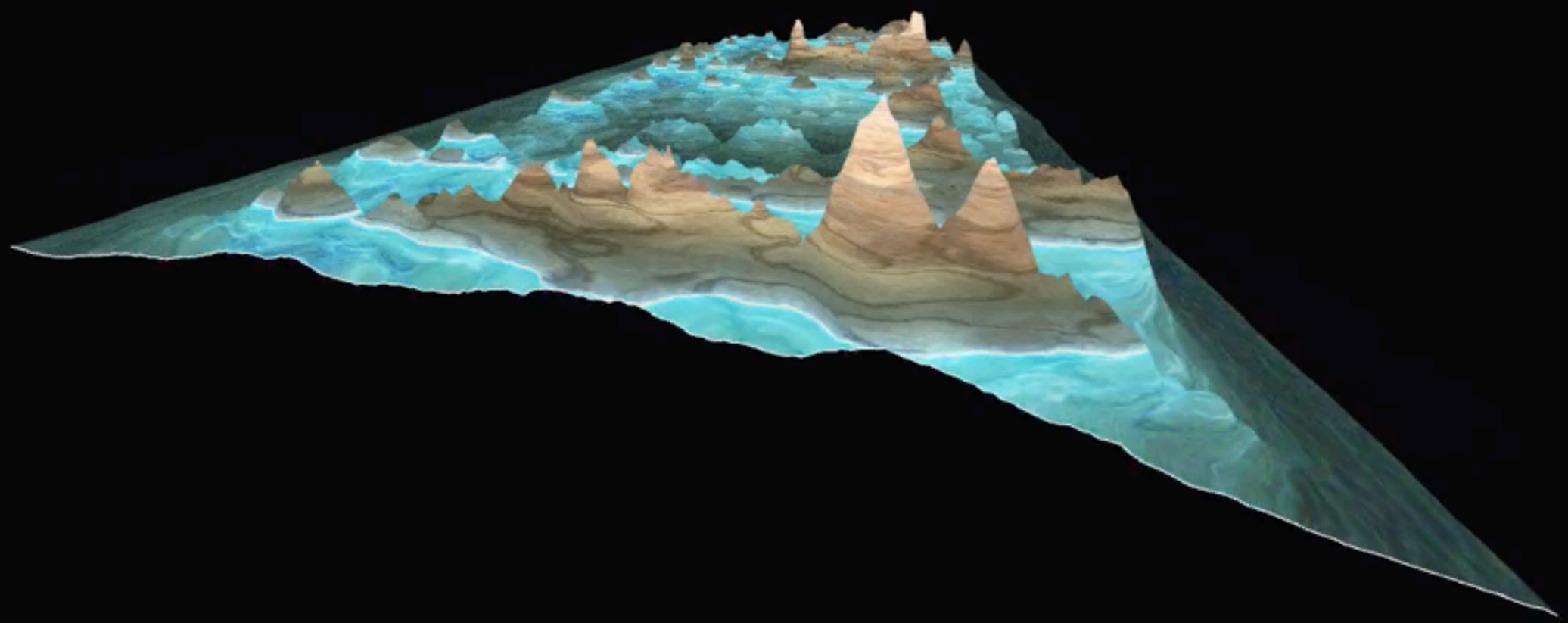
Lisbon's slow traffic areas

Each trail constitutes a temporary route where the average speed is mapped to its color. Pure green represents average speeds of 60 km/h. Cooler and greenish hues traduce rapid transit arteries, while the sluggish ones are reddish and hotter. There is a visual emphasis on the slower areas, with hot colors traducing sluggish traffic.

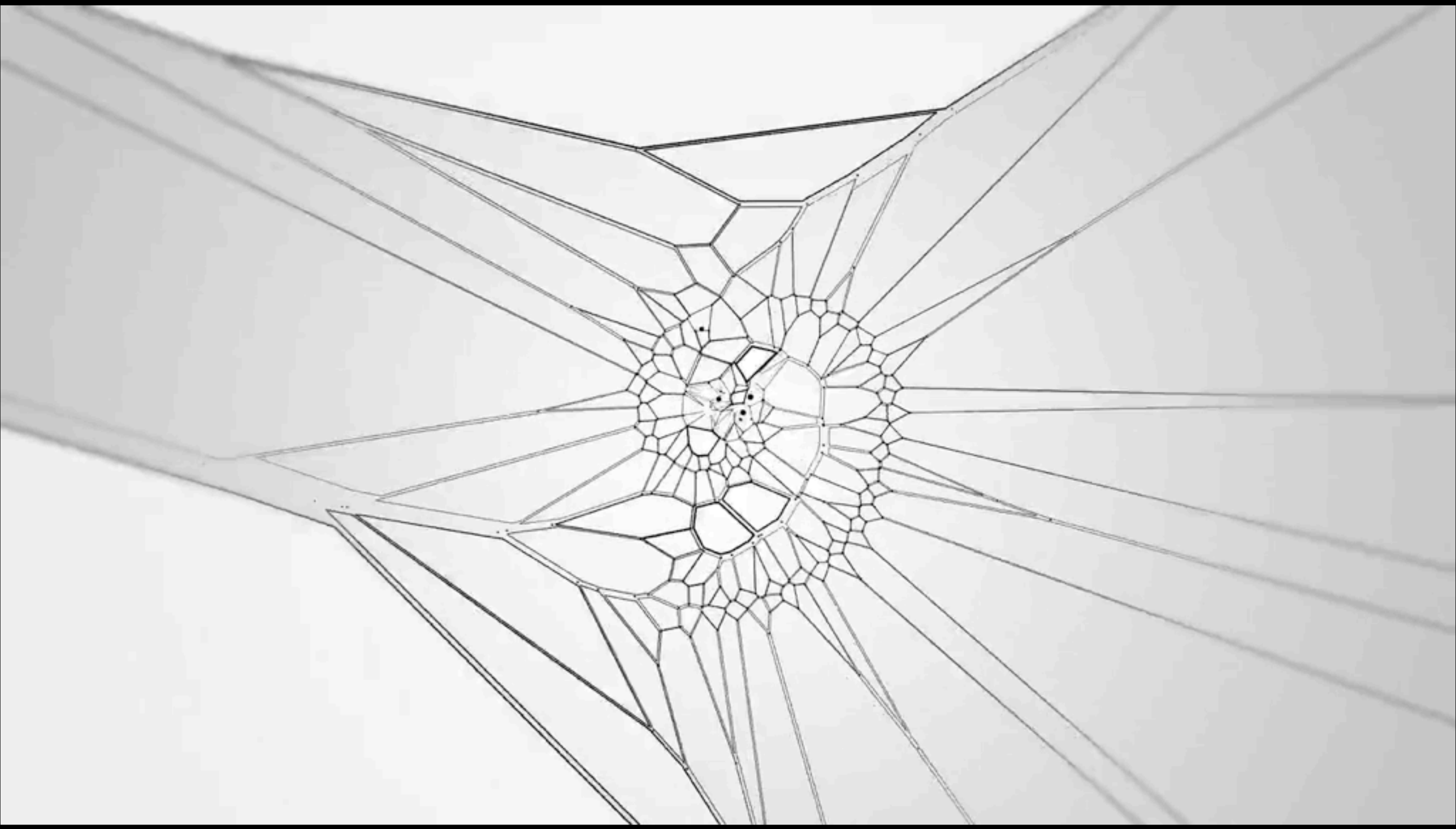
Lisbon's traffic speed and intensity

Each trail constitutes a temporary route where the average speed is mapped to its color. Pure green represents average speeds of 50 km/h. Therefore cooler and greenish hues traduce rapid transit arteries, while the sluggish ones are reddish and hotter. With this approach it is possible to visualize the evolution of the traffic speed and intensity in the main arteries and areas of the city.

TRAFFIC IN LISBON AND THE VISUALIZATION OF TRAJECTORIES
PEDRO M. CRUZ / 2010



AUDIO-GENERATED LANDSCAPE
ROBERT HODGIN / 2008



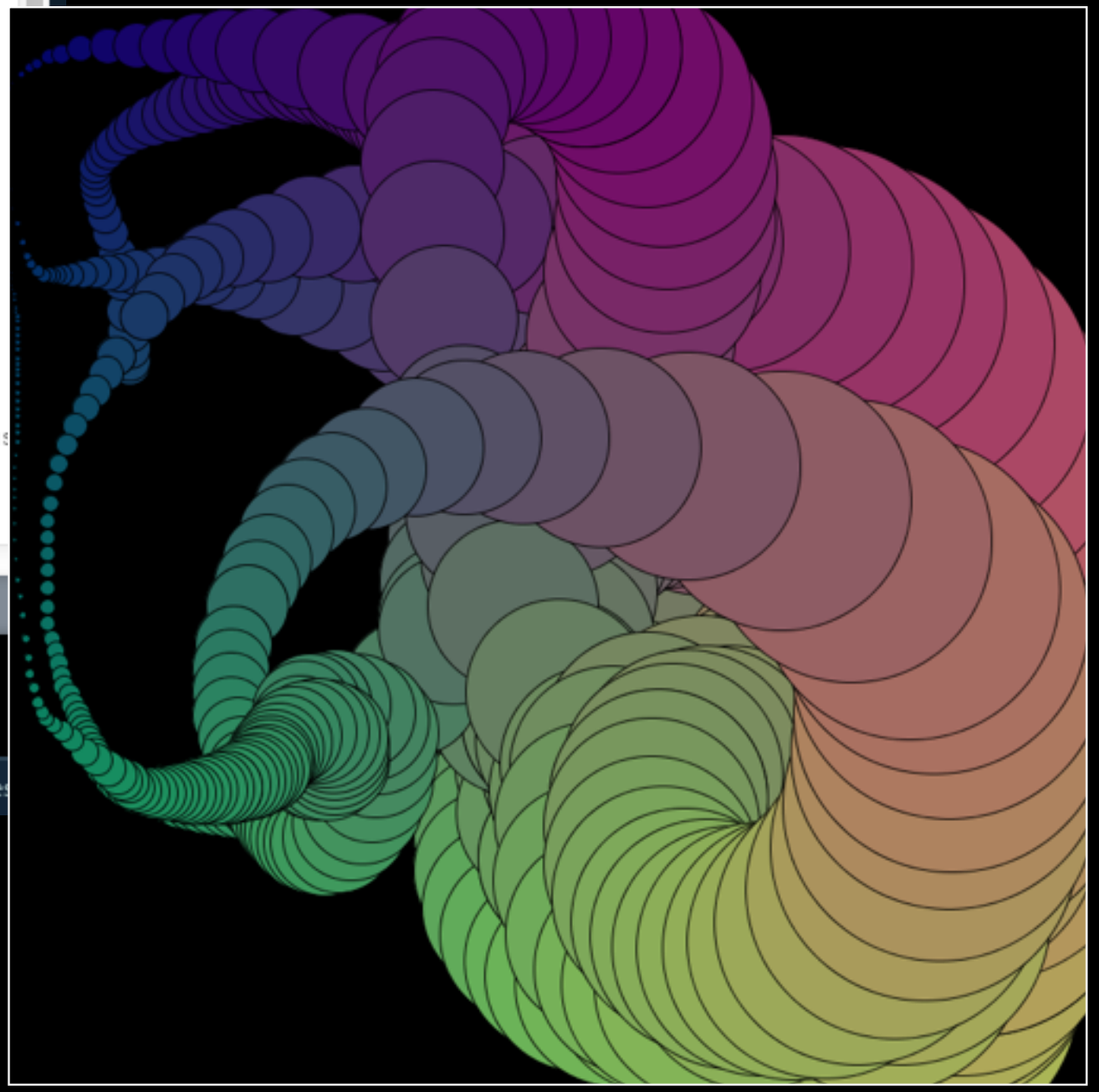
AUDIO-GENERATED LANDSCAPE
ROBERT HODGIN / 2008

Java ▾

LSI_012_mouseDrawing

```
1
2  // Starts the Scketch
3  void setup() {
4    //screen size
5    size(800, 800);
6    // painting the back in black
7    background(0);
8  }
9
10 // Looping forever
11 void draw() {
12
13   // border line color
14   stroke(0);
15   // fill color.
16   fill(mouseX/4, mouseY/4,100);
17
18  // Drawing ellipse on the same position as the mous
19   ellipse(mouseX, mouseY, mouseX/3, mouseX/3);
20 }
21
```

Console | Errors | Updates

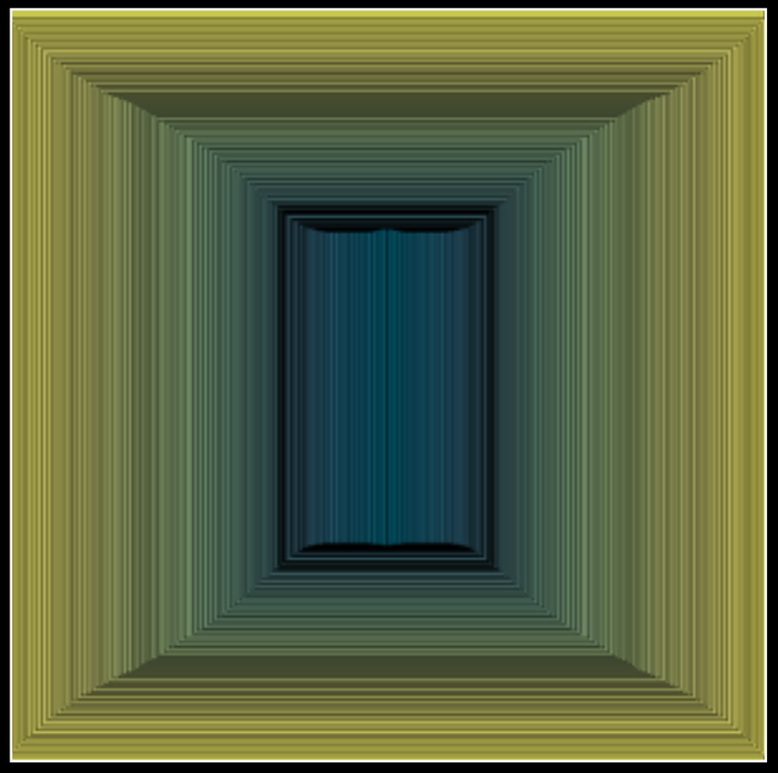


 # CRIA UM NOVO PROGRAMA SEMELHANTE AO DO SLIDE ANTERIOR.

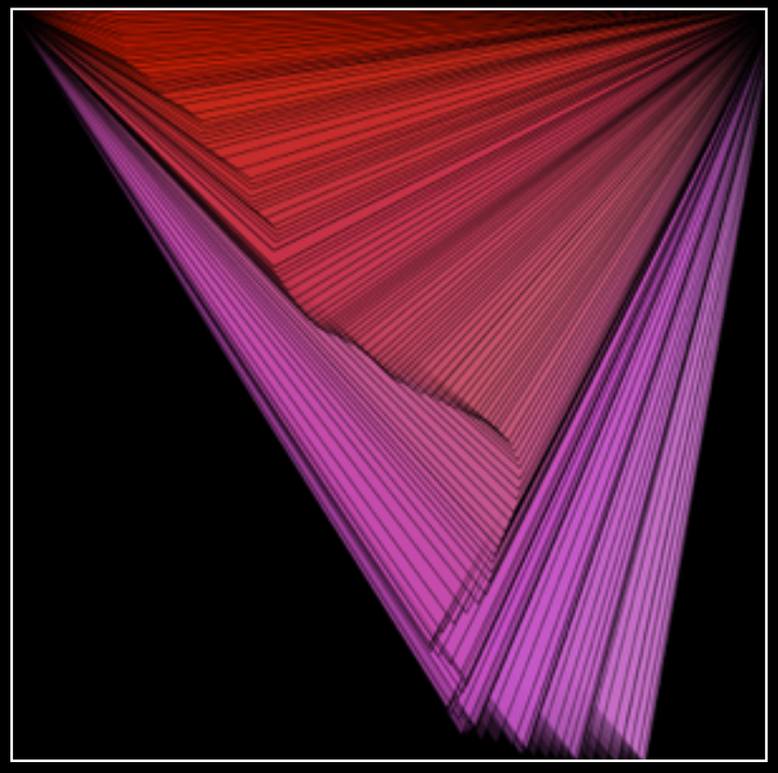
UTILIZANDO DIFERENTES FORMAS:
LINES(),TRIANGLE(), RECT(),…

CRIANDO DINAMISMOS COM CORES, DIMENSOES / POSIÇÃO MOUSE

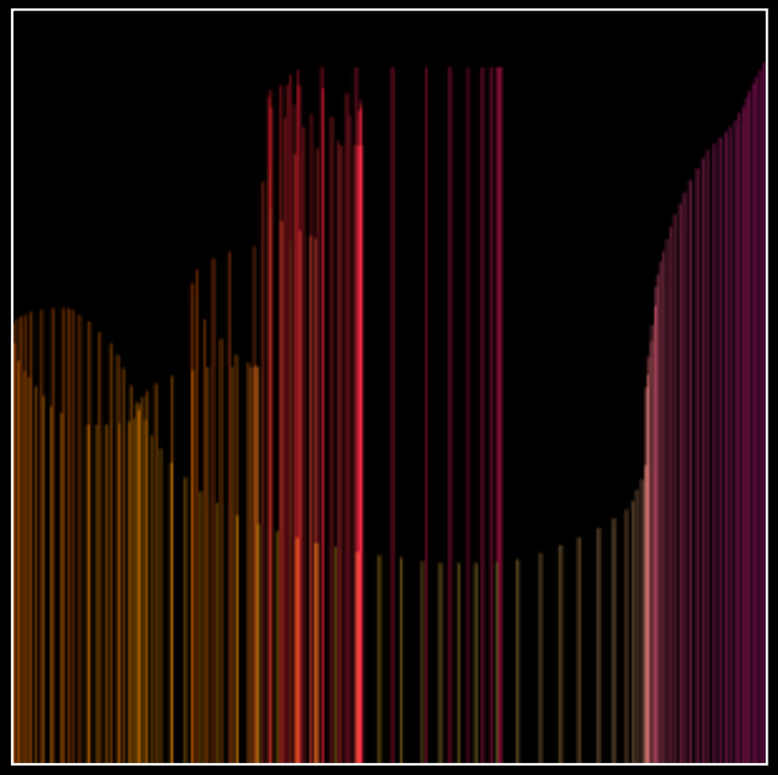
+ SAVEFRAME()



rect(, , ,)



triangle(, , ,)



line(, , ,)

```
// VARIABLE
int - integer number: 1,2,3,…
float - decimal numbers: 1.25, …
boolean - true / false
Ex:
int color = 255;
float posX= 20,5;
boolean active = false;


// CONDITIONALS

[Se isto acontecer -> faz isto]
if ( condition ) {

something happens. . .

}

else { . . .  }
[Caso contrario -> faz isto]
```

```
// OPERATORS
==  (equality): posX=0;
!= (inequality): posX!=0;
< (less than): posX<0;
> (greater than): pos>0;
<= (less or equal): pos<=0;
>= (greater or equal): pos>=0;


// LOGIC OPERATORS

[posX maior que 0 E menor que width]
&& (and): posX > 0 && posX <width

[posX menor que 0 OU maior que width]
|| (or): posX < 0 && posX >width

[mouse nao está a ser clicado]
! (not) !mousePressed
```

```
// EXAMPLES

[se a posiçãoX for inferior a 0px,
a posição passa a ser 800px]

if (posX < 0) {
posX = 800;
}


[se o Mouse for clicado a cor é 255,
caso contrario a cor é preta]

if (mousePressed==true) {
fill=255;
}
else {
fill=0;
}


[SE mouseX esta no primeiro terço do ecran background PRETO,
caso contrario, SE esta no meio do ecran BackgroundCinza,
caso contrario (lado direito do ecran), background branco]

if (mouseX < width/3) { background=0;}

else if (mouseX >= width/3 && mouseX <= (width/3)*2)
{background=150;}

else { background=255; }
```

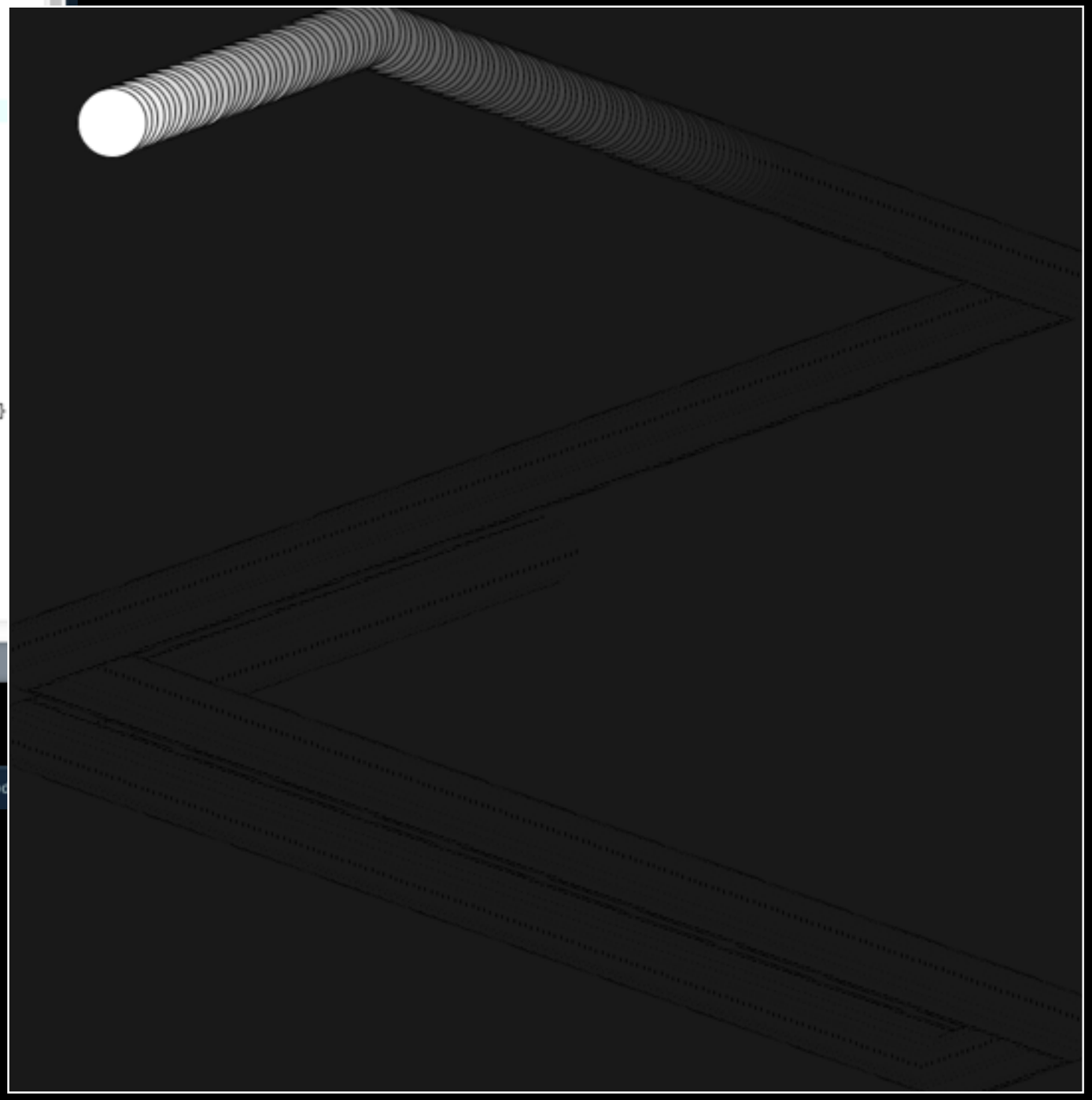
```
// EXAMPLES

[se a posiçãoX for inferior a 0px,
ou maior que o WIDTH,
o valor da velocidade inverte / negativo]

if (posX < 0 || posX > width) {
velX= -velX;
}


[se o Mouse NAO for clicado, a posiçao X
aumenta no valor de 1px]

if (!mousePressed) {
posX=posX+1;
}
```

Java ▾

LSI_016_circleBouncing ▾

```
5
6  // variable to store circles velocities X/Y
7  float velX=random(-5,5);
8  float velY=random(-5,5);
9
10 // variable to store the size of the ball
11 float ballSize = 80;
12
13 void setup (){
14 size(800,800); // Display dimensions
15 }
16
17
18 void draw (){
19
20 background(0);
21
22  // circles color
23  fill(255);
24
25 // if circle goes out of the screen, velocity is inverted
26 if(posX<(0+ballSize/2) || posX > (width-ballSize/2) ) {velX=-velX;}
27 if(posY<(0+ballSize/2) || posY > (height-ballSize/2) ) {velY=-velY;}
28
29 posX=posX+velX;
30 posY=posY+velY;
31
32  //drawing circle
33  ellipse(posX,posY, ballSize,ballSize);
34
35 }
```

Console Errors Upd



 # CRIA UM NOVO PROGRAMA SEMELHANTE AOS DO SLIDE ANTERIOR.

CRIANDO NOVOS DINAMISMOS COM *CORES*, *DIMENSÕES / POSIÇÃO MOUSE*

DIFERENTES OBSTÁCULOS

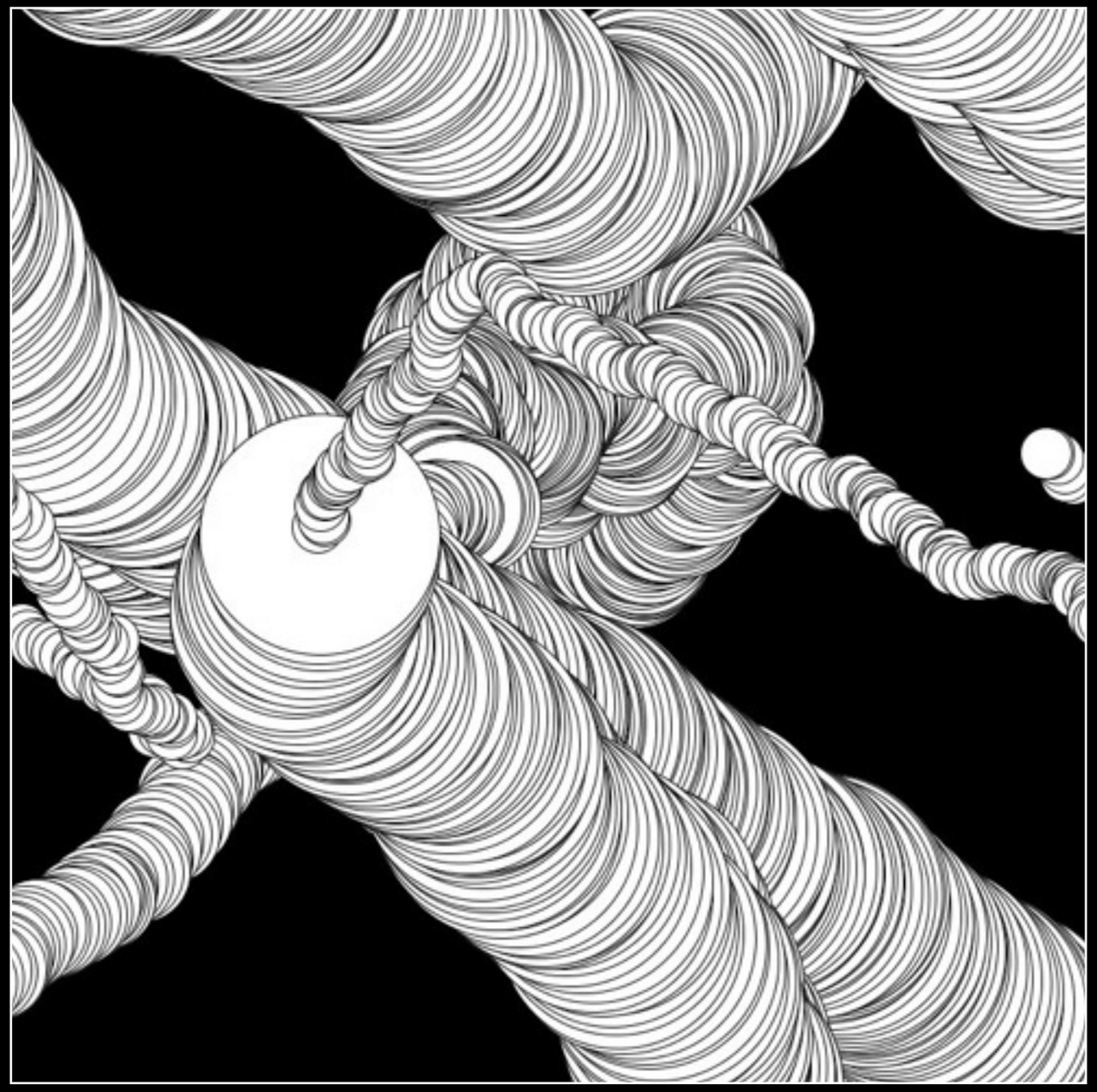
VELOCIDADES VARIAVEIS

INTERACÇÕES COM CLICKS EM RATO / TECLADO

BACKGROUND/NO BACKGROUND

RANDOM()

GRAVAR SEQUÊNCIA DE FRAMES PARA EDITAR UM SAVEFRAME()



```
// MOUSE CLICK
void mouseClicked() {
..something happens here..
}

void mouseDragged() { . . . }

void mouseReleased() { . . . }

// KEYBOARD
void keyReleased() { . . . }

// SAVING PICTURE
saveFrame("filename-#####.jpg");

// SCALING A NUMBER (REGRA 3 SIMPLES)
map(value, start1, stop1, start2, stop2)
```