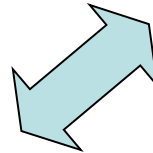
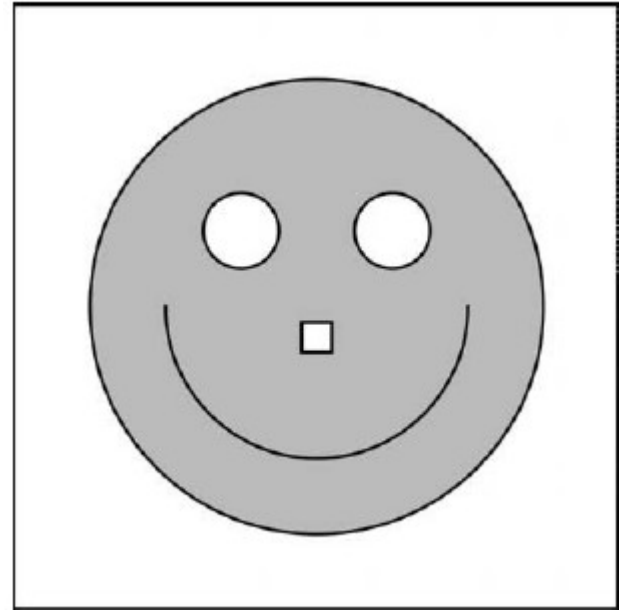
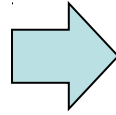


# Programação para Artistas com Processing

Profa. Rosilane Mota

Prof. Wallace Lages

# Porque Processing?

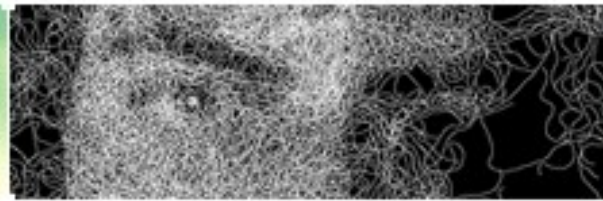


# Porque Processing?

- Programar com imagens, animações e som
- Aprender, prototipar e produzir
- Feedback visual imediato!
- Open source, baseado em Java



[Silica-Esc](#)  
by Vladimir Todorovic



[Mycelium](#)  
by Ryan Alexander



[The Jelly Face Experiments](#)  
by NYX (Liina Nilsson)



[New Year's Cards, 2001-2010](#)  
by Golan Levin



[COP15 Generative Identity](#)  
by Mikkel Koser



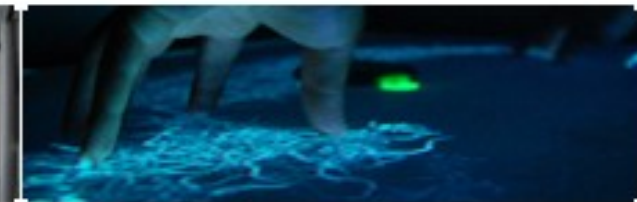
[Branching Morphogenesis](#)  
by Sabin+Jones LabStudio



[Big Screens 2](#)  
by NYU ITP

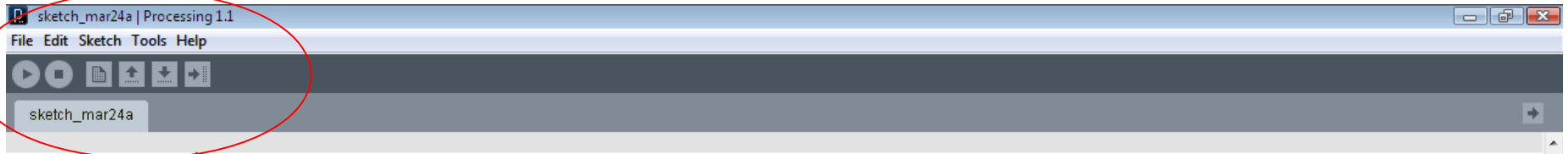


[The Emergence Project](#)  
by Daniel Sauter, Mark Hereld



[Roots](#)  
by Memo Akten, Owen Vallis,  
Jordan Hochenbaum

# Processing

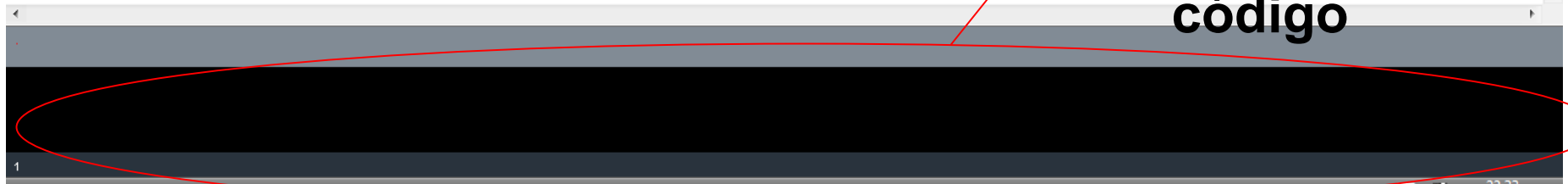


**Controles**

**Código**



**Mensagens  
de  
erro do  
código**



# Desenhando com processing

```
rect(10,10,10,20);
```

# Desenhando com processing

```
rect(x,y,largura,altura);
```

Letras minúsculas



Indica o fim do comando

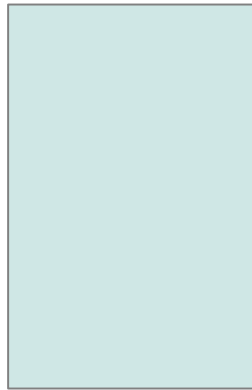


# Aumentando a área ....

```
size(800,600);
```

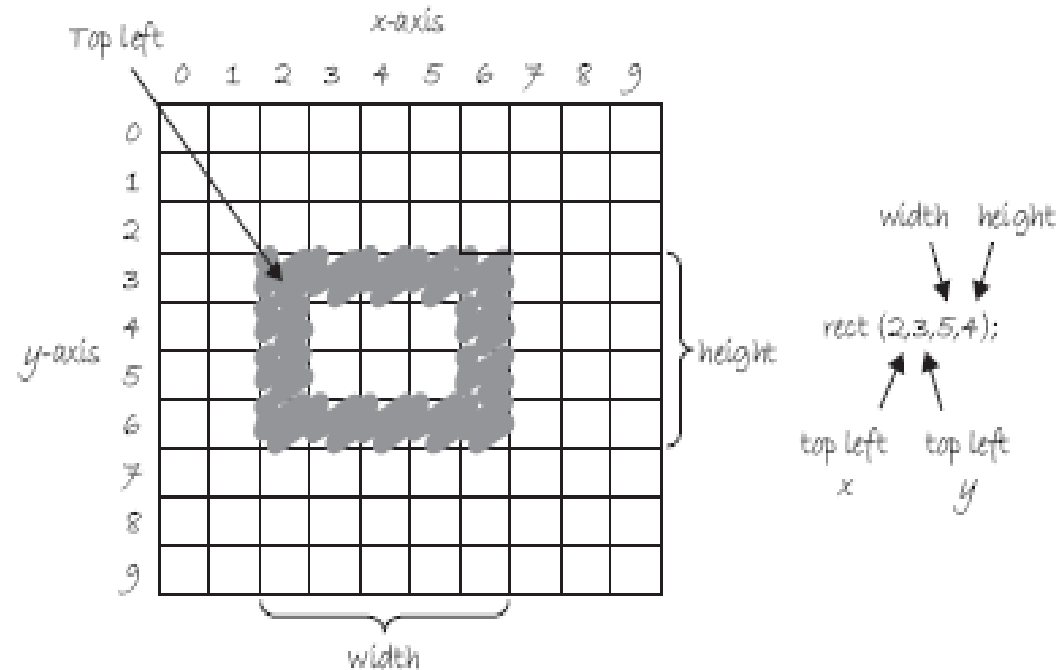


# Desenhando com processing



# Sistema de coordenadas:

```
size(800,600);  
background(0,0,0);  
rect(0,0,100,200);
```



```
// Canto superior esq (0,0) e também é para a janela!  
// Largura 100, Altura 200
```

# Mais opções ...

Mude a cor do fundo:

```
background(r,g,b);
```

r,g,b pode variar de 0 a 255

# Mais opções ...

Mude a cor da linha:

```
stroke(r,g,b);
```

r,g,b pode variar de 0 a 255

# Mais opções ...

Mude a cor do retângulo:

```
fill(r,g,b);
```

r, g, b podem variar de 0 a 255

# Mais opções ...

`size(largura,altura)`

`rect(x,y,largura,altura);`

`fill(r,g,b);`

`stroke(r,g,b);`

`background(r,g,b);`

# Mais opções ...

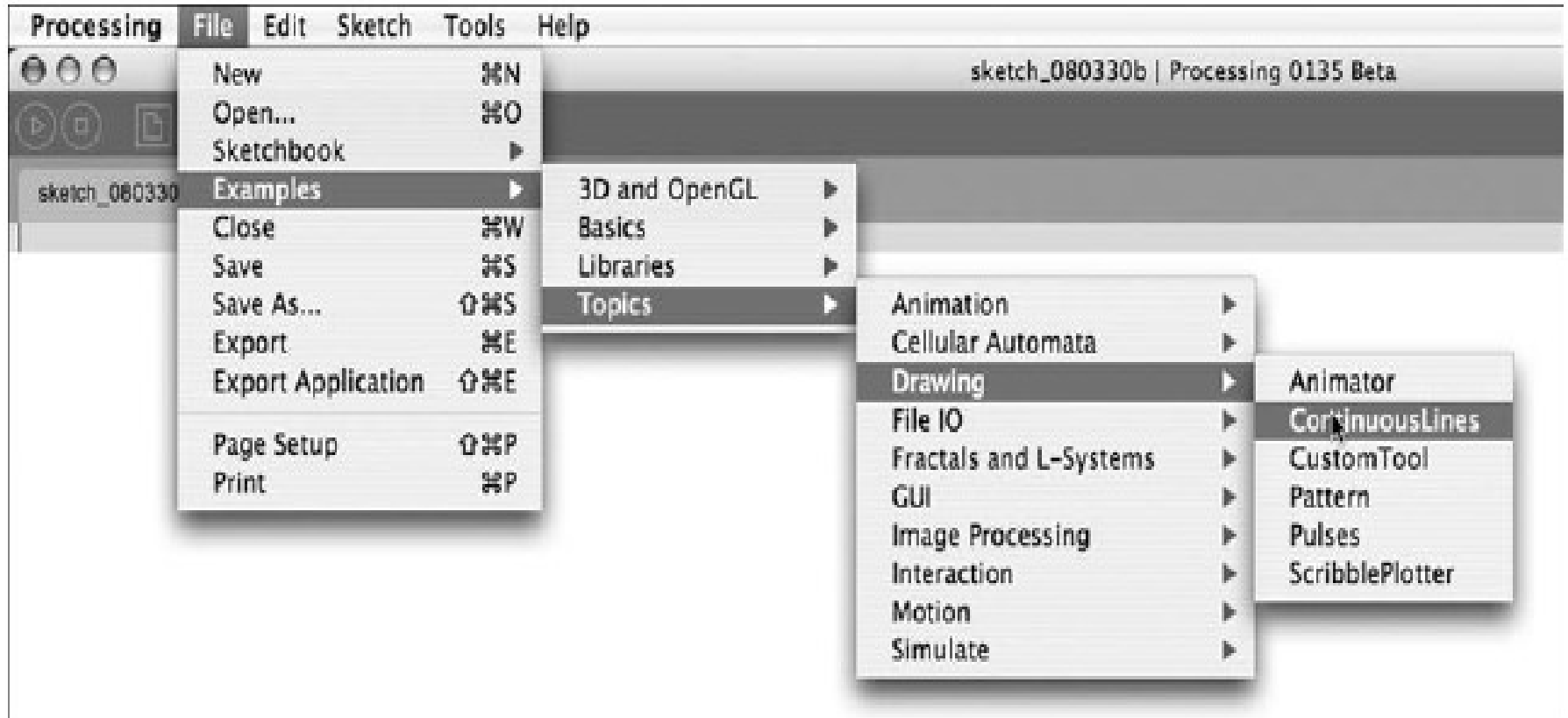
- Menu Tools->Color Selector

# Transparência...

- Acrescente transparência (canal alpha)  
→ as mesmas funções com mais um parâmetro: (r, g, b, a)



# Processing



# Exemplo

```
size(200,200);  
background(0);  
noStroke();
```

```
// 75% opaco  
fill(255,0,0,191);  
rect(0,50,200,40);
```

```
// Padrão: 100% opaco  
fill(0,0,255);  
rect(0,0,100,200);
```

```
// 55% opaco  
fill(255,0,0,127);  
rect(0,100,200,40);
```

```
// 255 é 100% opaco  
fill(255,0,0,255);  
rect(0,0,200,40);
```

```
// 25% opaco  
fill(255,0,0,63);  
rect(0,150,200,40);
```

# Outras figuras – Elipse

// Exemplo figuras geométricas

// Autora: Rosilane Mota

size(400,400);

background(255);

stroke(0);

fill(#0000FF); // Azul

ellipse(200,200,100,200);

// Centro (0,0)

// Largura 100, Altura 200

# Outras figuras – Arco

// Exemplo figuras geométricas

// Autora: Rosilane Mota

size(400,400);

background(255);

stroke(0);

fill(#0000FF); // Azul

arc(200,200,300,150,0,PI);

// Centro (200,200)

// Largura 300, Altura 150

// Começa a desenhar em 0 radianos e termina em PI radianos (0 a  
// 180 graus)

# Outras figuras – Linha

// Exemplo figuras geométricas

// Autora: Rosilane Mota

```
size(400,400);
```

```
background(255);
```

```
stroke(0);
```

```
fill(#0000FF); // Azul
```

```
line(0,0,200,200);
```

// Começa a linha no (0,0) e termina no (200,200)

# Smile!

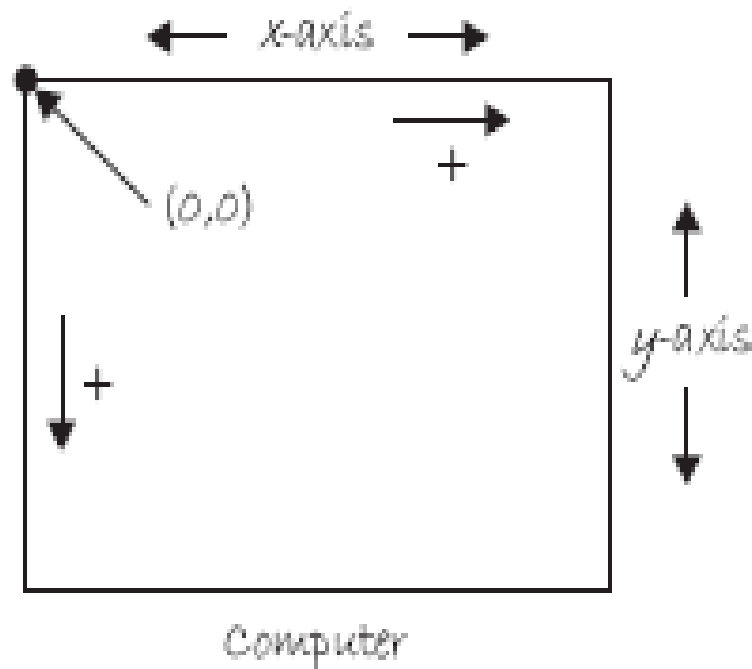
```
// Exemplo figuras geométricas  
// Autora: Rosilane Mota
```

```
size(400,400);  
background(255);  
stroke(0);  
fill(150);  
ellipse (200,200,350,350);  
fill(255);  
rect(110,110,50,50);  
rect(240,110,50,50);  
fill(#ff0000);  
ellipse (200,200,50,50);  
noFill();  
arc(200,200,200,200,0,PI);
```

# Exercício 1

- Crie seu desenho com as figuras geométricas aprendidas e as cores que desejar

- Lembre-se!



# Talvez facilite

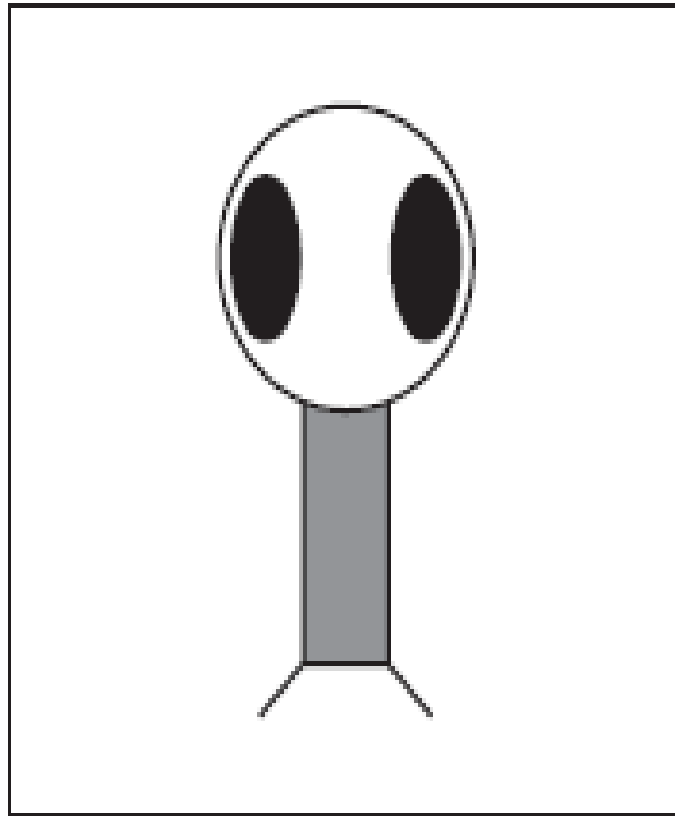
```
rectMode(CENTER);
```

Ponto de referência é o centro da figura,  
como na elipse.



# Exercício 2

- Crie o código para o desenho do Zoog abaixo



# Faça ele “falar”

- Acrescente ao final do código o comando  
`println(“Leve-me ao seu líder!”);`

# O código ficou assim?

```
size(200,200);  
background(255);  
smooth();
```

```
ellipseMode(CENTER);  
rectMode(CENTER);
```

```
// Corpo  
stroke(0);  
fill(150);  
rect(100,100,20,100);
```

```
// Cabeça  
fill(255);  
ellipse(100,70,60,60);
```

```
// Olhos  
fill(0);  
ellipse(81,70,16,32);  
ellipse(119,70,16,32);
```

```
// Pernas  
stroke(0);  
line(90,150,80,160);  
line(110,150,120,160);
```

```
println("Leve-me ao seu líder!");
```

# Mensagens de Erro

- Retire um ponto e vírgula e execute novamente.
- Observe a mensagem de erro e a indicação da linha em destaque no código.
- Coloque novamente o ponto e vírguloa.
- Altere a primeira letra de algum comando para maiúscula e observe a mensagem de erro.
- Corrija novamente.

# Comentários

- Uma linha

// Exemplo uma linha

- Mais de uma linha

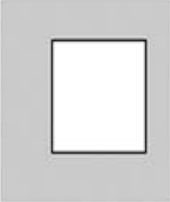
```
/* Exemplo  
   comentário  
   mais de uma linha  
*/
```

rect() \ Language (API) \ Processing 1.0 (BETA)

http://processing.org/reference/rect\_.html

rect() \ Language (API) \ Pr...

Name **rect()**

Examples  `rect(30, 20, 55, 55);`

Description Draws a rectangle to the screen. A rectangle is a four-sided shape with every angle at ninety degrees. The first two parameters set the location, the third sets the width, and the fourth sets the height. The origin is changed with the **rectMode()** function.

Syntax `rect(x, y, width, height)`

Parameters

x	int or float: x-coordinate of the rectangle
y	int or float: y-coordinate of the rectangle
width	int or float: width of the rectangle
height	int or float: height of the rectangle

Usage Web & Application

Related [rectMode\(\)](#)  
[quad\(\)](#)

# Gere sua aplicação

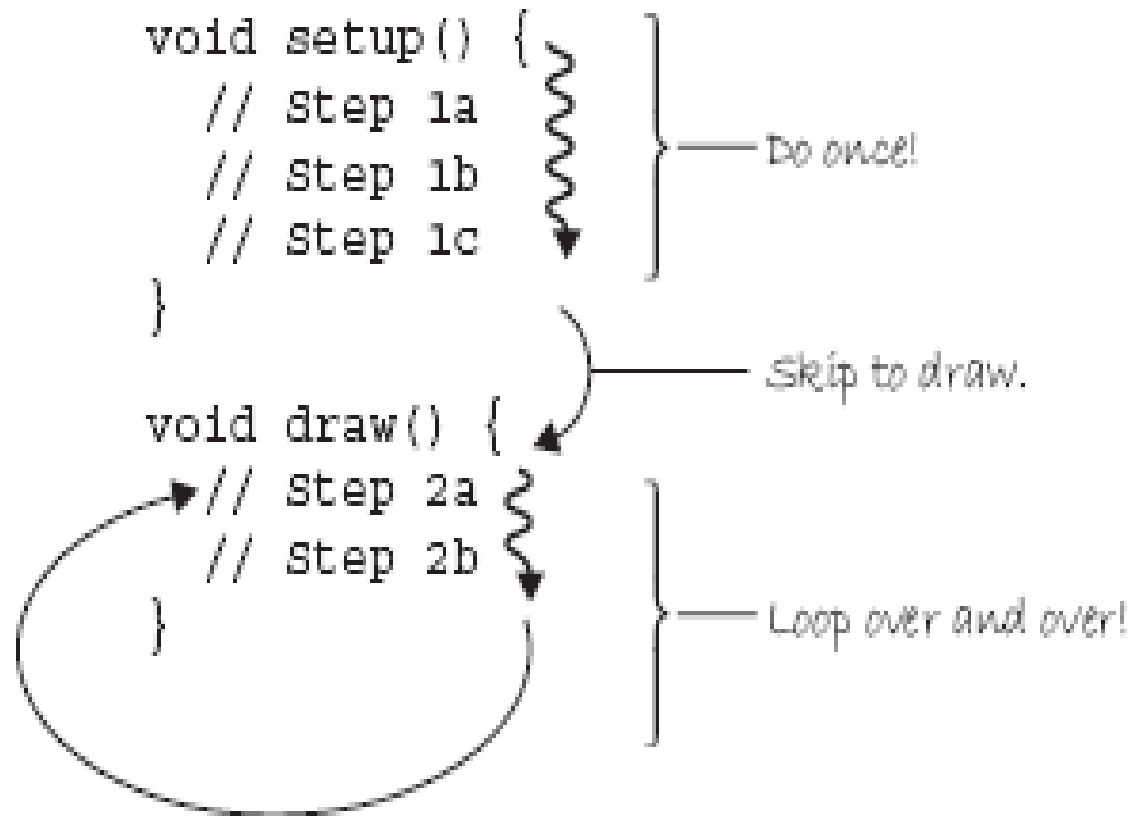
- Gere o executável
  - File->Export Application
- Gere o applet
  - File->Export

# Alguns conceitos

- setup
  - Configurações iniciais: tamanho da janela, cor do fundo
  - Ele é executado só no início
- draw
  - O que é desenhado e atualizado a cada momento: posição do personagem na tela, interação do espectador
  - Ele é executado a cada atualização da tela do monitor



# Alguns conceitos



# Aplique no código

```
void setup()
{
  size(200,200);
}
```

```
void draw()
{
  background(255);
  smooth();

  ellipseMode(CENTER);
  rectMode(CENTER);
```

```
// Corpo
stroke(0);
fill(150);
rect(100,100,20,100);
```

```
// Cabeça
fill(255);
ellipse(100,70,60,60);
```

```
// Olhos
fill(0);
ellipse(81,70,16,32);
ellipse(119,70,16,32);
```

```
// Pernas
stroke(0);
line(90,150,80,160);
line(110,150,120,160);
```

```
println("Leve-me ao seu líder!");
}
```

# Verificando Informações do Mouse

```
void setup()  
{  
  size(200,200);  
}
```

```
void draw()  
{  
  background(255);
```

```
  stroke(0);  
  fill(175);  
  rectMode(CENTER);  
  rect(mouseX,mouseY,50,50);  
}
```

# Verificando Informações do Mouse

- Mude o comando background para a parte do Setup
- O que aconteceu?
- Porque?

# Altere o código...

```
void setup()
{
  size(200,200);
}

void draw()
{
  background(255);
  smooth();

  ellipseMode(CENTER);
  rectMode(CENTER);

  // Corpo
  stroke(0);
  fill(150);
  rect(mouseX,mouseY,20,100);

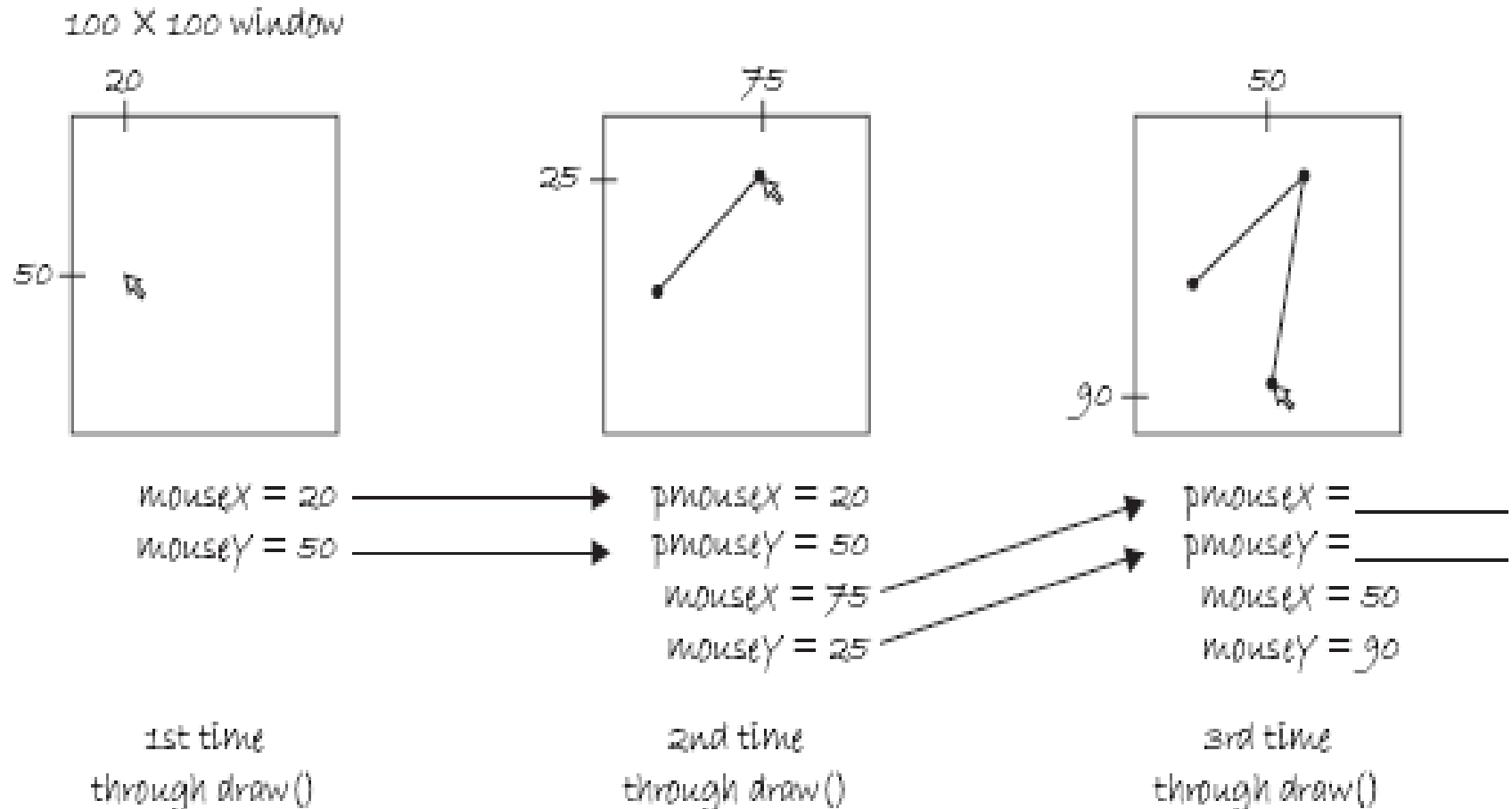
  // Cabeça
  fill(255);
  ellipse(mouseX,mouseY,60,60);

  // Olhos
  fill(0);
  ellipse(81,70,16,32);
  ellipse(119,70,16,32);

  // Pernas
  stroke(0);
  line(90,150,80,160);
  line(110,150,120,160);

  println("Leve-me para o seu líder!");
}
```

# Mais informações do Mouse



# Desenhe uma linha contínua

```
void setup()  
{  
  size(200,200);  
  background(255);  
  smooth();  
}
```

```
void draw()  
{  
  stroke(0);  
  line(mouseX , mouseY , mouseX , mouseY );  
}
```

# Altere a espessura da linha

```
void setup()  
{  
  size(200,200);  
  background(255);  
  smooth();  
}
```

```
void draw()  
{  
  stroke(0);  
  strokeWeight(5);  
  line(pmouseX ,pmouseY ,mouseX ,mouseY );  
}
```



# Clique do Mouse e tecla pressionada

```
void setup()  
{  
  size(200,200);  
  background(255);  
}
```

```
void draw()  
{  
}
```

```
void mousePressed()  
{  
  stroke(0);  
  fill(175);  
  rectMode(CENTER);  
  rect(mouseX,mouseY,16,16);  
}
```

```
void keyPressed()  
{  
  background(255);  
}
```

# Zoog Interativo!

```
void setup()
{
  size(200,200);
  smooth();
  frameRate(30) ;
}
```

```
void draw()
{
  background(255);
```

```
  ellipseMode(CENTER);
  rectMode(CENTER);
```

```
  // Corpo
  stroke(0);
  fill(175);
  rect (mouseX, mouseY, 20, 100);
```

```
  // Cabeça
  stroke(0);
  fill(255);
  ellipse (mouseX ,mouseY-30, 60, 60);
```

```
// Olhos
fill(mouseX, 0, mouseY);
ellipse(mouseX-19, mouseY-30, 16, 32);
ellipse(mouseX + 19, mouseY-30, 16, 32);

// Pernas
stroke(0);
line(mouseX-10, mouseY + 50, pmouseX-
      10, pmouseY + 60);
line(mouseX + 10, mouseY + 50,
      pmouseX + 10, pmouseY + 60);
}

void mousePressed()
{
  println( "Leve-me para o seu
           líder!! ");
}
```

# Crie seu próprio projeto!

- Use interação com mouse...
- Com teclado...
- E se divirta!

