

Programação para Artistas com Processing

Profa. Rosilane Mota

Prof. Wallace Lages

Aula 3

- Revisão Condicionais
- Loops
- Funções

Condicionais

Expressões booleanas – Operadores relacionais

> maior que

< menor que

>= maior ou igual

<= menor ou igual

== igual

!= diferente

Condicionais

Expressões booleanas – if, else, else if

Se o mouse estiver no lado esquerdo da tela

Desenhe um retângulo do lado esquerdo da tela

```
if( mouseX < width/2){  
    Fill(255)  
    rect(0,0,width/2,width/2)  
}
```

Condicionais

Operadores lógicos

Se o mouse estiver no lado direito da tela

E

o mouse estiver na parte de baixo da tela

Desenhe um retângulo no canto direito inferior

```
if (mouseX > width/2) {  
    if (mouseY > height/2) {  
        fill(255);  
        rect(width/2,height/2,width/2,height/2);  
    }  
}
```

Condicionais

Operadores lógicos

E lógico \rightarrow (Expressão A) && (Expressão B)

Ou Lógico \rightarrow (Expressão A) || (Expressão B)

Não Lógico \rightarrow !(Expressão A)

Condicionais

Operadores lógicos

```
if (mouseX > width/2) {  
    if (mouseY > height/2) {  
        fill(255);  
        rect(width/2,height/2,width/2,height/2);  
    }  
}  
  
if((mouseX > width/2) && (mouseY > height/2)) {  
    fill(255);  
    rect(width/2,height/2,width/2,height/2);  
}
```

Condicionais

Operadores lógicos

```
if (!mousePressed) {  
    ellipse(width/2,height/2,100,100);  
} else {  
    rect(width/2,height/2,100,100);  
}
```


Condicionais

Operadores lógicos

As expressões seguintes são verdadeiras?

Suponha $x=5$ e $y=6$

`!(x > 6)`

`(x == 6 && x == 5)`

`(x == 6 || x == 5)`

`(x > -1 && y < 10)`

Condicionais

Operadores lógicos

As expressões seguintes são verdadeiras?

Suponha $x=5$ e $y=6$

`!(x > 6)`

`(x == 6 && x == 5)`

`(x == 6 || x == 5)`

`(x > -1 && y < 10)`

Condicionais

Operadores lógicos - Exercício

Escreva um programa que:

- 1-Tenha uma tela 200x200 dividida em 4 quadrantes
- 2-Acenda o quadrante no qual o mouse esteja localizado

Condicionais

Expressões booleanas – exercício

```
void draw() {  
  
    background(255);  
    stroke(0);  
    line(100, 0, 100, 200);  
    line(0, 100, 200, 100);  
    noStroke();  
    Fill(0);  
  
    if (mouseX < 100 && mouseY < 100) {  
        rect(0, 0, 100, 100);  
    } else if (mouseX > 100 && mouseY < 100) {  
        rect(100, 0, 100, 100);  
    } else if (mouseX < 100 && mouseY > 100) {  
        rect(0, 100, 100, 100);  
    } else if (mouseX > 100 && mouseY > 100) {  
        rect(100, 100, 100, 100);  
    }  
}
```

Condicionais

Exercício - Gravidade!

$$\text{Posição Atual} = \text{Posição anterior} + \dot{\Delta} \text{Posição}$$
$$\dot{\Delta} \text{Posição} = \text{Velocidade} * \dot{\Delta} \text{Tempo}$$

Condicionais

Exercício - Gravidade!

$$\text{Posição Atual} = \text{Posição anterior} + \Delta \text{Posição}$$
$$\Delta \text{Posição} = \text{Velocidade} * \Delta \text{Tempo}$$

$$\text{Velocidade} = \text{Velocidade anterior} + \Delta \text{Velocidade}$$
$$\Delta \text{Velocidade} = \text{Aceleração} * \Delta \text{Tempo}$$

$$\Delta \text{Tempo} = \text{Tempo entre duas chamadas draw()}$$

`pos = old_pos + speed`

`speed = old_speed + acc`

Condicionais

Exercício - Gravidade!

Escreva um programa que:

1-Desenhe um quadrado da posição (100,0)

3-Inicie com velocidade zero

3-Caia com uma aceleração de 0.1 para baixo

4-Ao bater no fundo reverta sua velocidade para -0.95 da velocidade de impacto

$pos = old_pos + speed$

$speed = old_speed + acc$

Condicionais

Exercício - Gravidade!

```
float x = 100;  
float y = 0;  
float speed = 0;  
float gravity = 0.1;
```

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


Condicionais

Exercício - Gravidade!

```
void draw() {  
    //apaga o fundo  
    background(255);  
  
    // Desenha o quadrado  
    fill(0);  
    noStroke();  
    rectMode(CENTER);  
    rect(x, y, 10 , 10);  
}
```

Condicionais

Exercício - Gravidade!

```
y = y + speed;  
Speed = speed + gravity;  
  
// Reverte a velocidade  
if (y > height) {  
    y = height;  
    speed = speed * -0.95;  
}  
}
```

Laços de Repetição

- O que é?
 - Comandos agrupados que são executados de acordo com condição (igual IF)
 - Mas...
 - Se condição for falsa, os comandos são executados novamente! (Repete)

Laços de Repetição

- Quando usar?
 - Quando você perceber que está executando a mesma lógica mais de uma vez, com apenas as informações mudando (“variando”) para os mesmos comandos

Laços de Repetição

- Duas formas:
 - While
 - For

Laços de Repetição

While

Enquanto (Expressão for verdadeira) Faça

Comando 1

Comando 2

Comando 3

...

Comando N

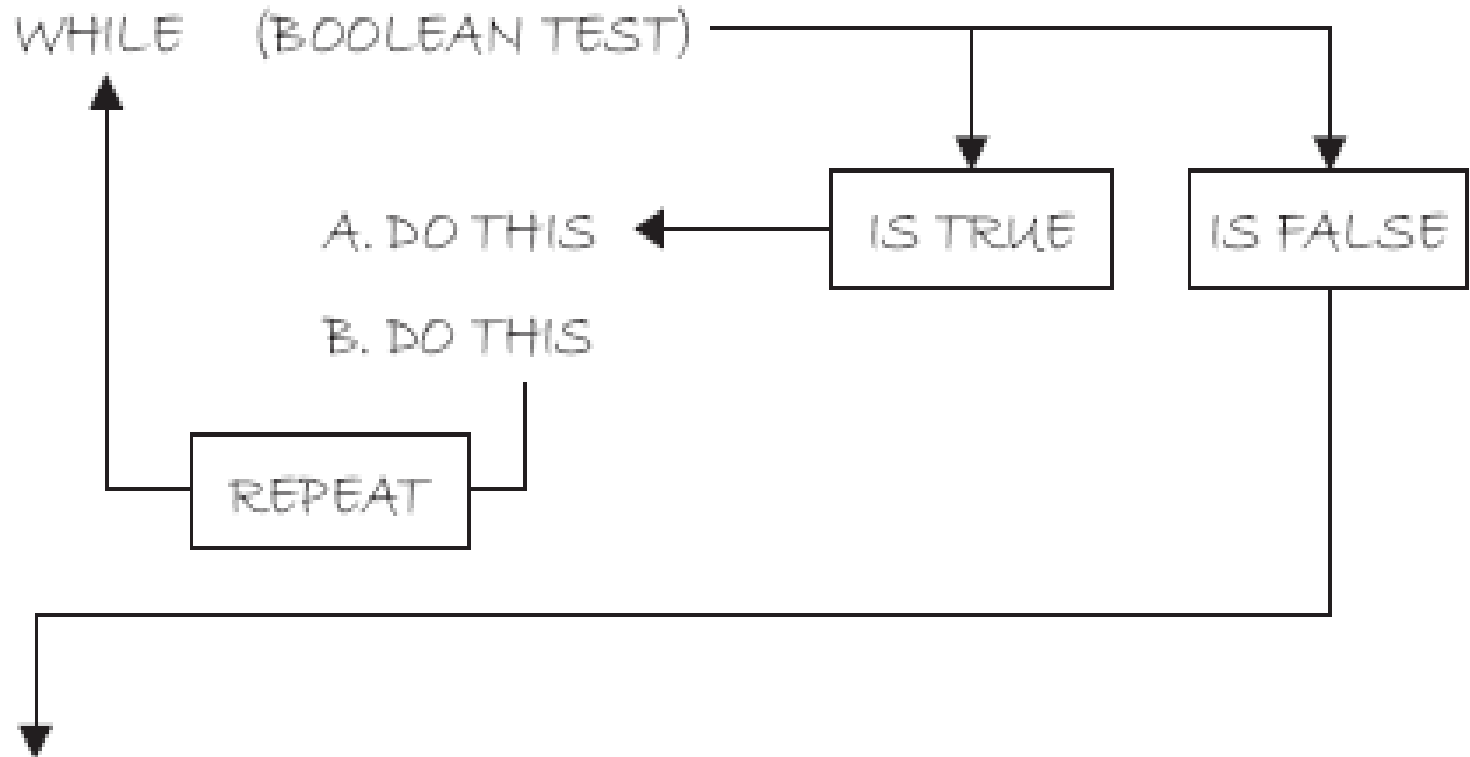
```
While (Expression)
```

```
{
```

```
}
```

Laços de Repetição

While



Laços de Repetição

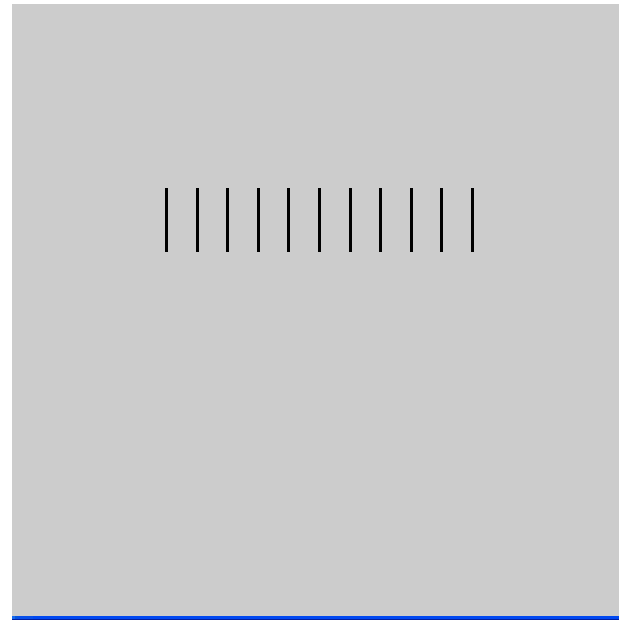
While

```
int i = 0;
while( i < 10)
{
    println(i);
    i = i + 1;
}
```


Laços de Repetição

While

```
size(200,200);  
Background(255);  
  
// Legs  
stroke(0);  
line(50,60,50,80);  
line(60,60,60,80);  
line(70,60,70,80);  
line(80,60,80,80);  
line(90,60,90,80);  
line(100,60,100,80);  
line(110,60,110,80);  
line(120,60,120,80);  
line(130,60,130,80);  
line(140,60,140,80);  
line(150,60,150,80);
```



Laços de Repetição

While

- Igual:
 - Todas começam na mesma altura $Y=60$
 - Todas possuem comprimento 20
 - O espaço entre as pernas é 10
- Variação:
 - A primeira perna começa na posição $X=50$
 - A última perna termina em 150

Laços de Repetição

While

- Mudando código...
 - O que é igual vira variável com o valor inicial

```
int y = 60;  
int altura = 20;
```

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

```
// Pernas do Zoog  
stroke(0);  
line(50,y,50,y+altura);  
line(60,y,60, y+altura);  
line(70,y,70, y+altura);  
line(80,y,80, y+altura);
```

```
line(90,y,90, y+altura);  
line(100,y,100, y+altura);  
line(110,y,110, y+altura);  
line(120,y,120, y+altura);  
line(130,y,130, y+altura);  
line(140,y,140, y+altura);  
line(150,y,150, y+altura);
```


Laços de Repetição

While

- Mudando código...

- O x começa no 50, termina no 150 e aumenta de 10 em 10

```
int x = 50, y = 60, altura = 20;
```

```
int espaco = 10;
```

```
size(200,200);
```

```
background(255);
```

```
// Pernas do Zoog
```

```
stroke(0);
```

```
while (x < 150)
```

```
{
```

```
    line(x,y,x,y+altura);
```

```
    x = x + espaco;
```

```
}
```

Mais fácil, não é ?

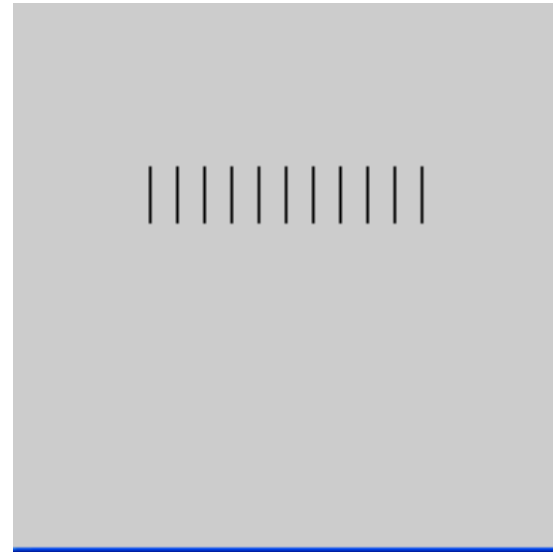
**Você prestou ATENÇÃO
na condição?**

Porquê não é <=, >, == ?

Laços de Repetição

While

```
int y = 60;  
int x = 50;  
int spacing = 10;  
int len = 20;  
int endLegs = 150;  
  
Stroke(0);  
  
while (x <= endLegs) {  
    line (x,y,x,y + len);  
    x = x + spacing;  
}
```



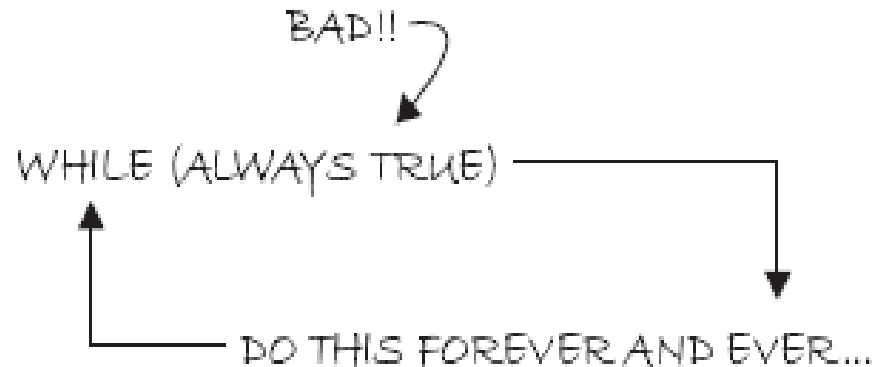
Laços de Repetição

While

- Quantas vezes ele repete os comandos abaixo?

```
int x = 0;
while (x < 10)
{
    println(x);
    x = x - 1;
}
```

Isso se chama **LOOP INFINITO!**



Por isso, tem que analisar bem qual condição será utilizada!

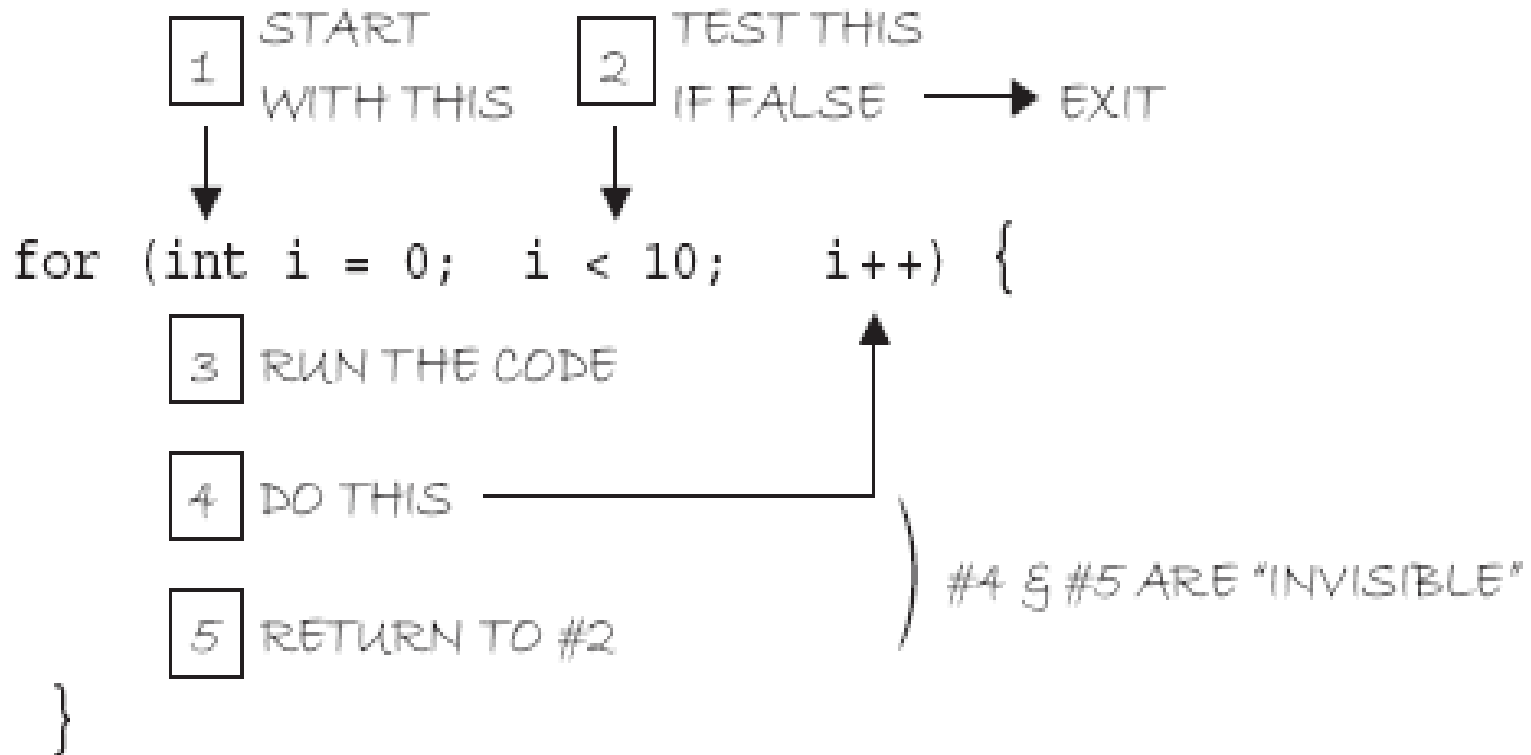
Laços de Repetição

For

```
Para variável = 10,  
    Enquanto variável < 20  
        Comando 1  
        Comando 2  
        Variável ++
```

```
for(i=10;i<20;i++)  
{  
    //comando  
}
```


Laços de Repetição



Laços de Repetição

For

```
int y = 60;  
int spacing = 10;  
int len = 20;  
  
for (int x = 50; x <= 150; x += spacing)  
{  
    line(x, y, x, y + len);  
}
```

Laços de Repetição

Comparando While e For

```
int x = 50, y = 60, altura = 20;  
int espaco = 10;
```

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

```
// Pernas do Zoog  
stroke(0);
```

```
while (x < 150)  
{  
    line(x,y,x,y+altura);  
    x = x + espaco;  
}
```

```
int y = 60, altura = 20;  
int espaco = 10;
```

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

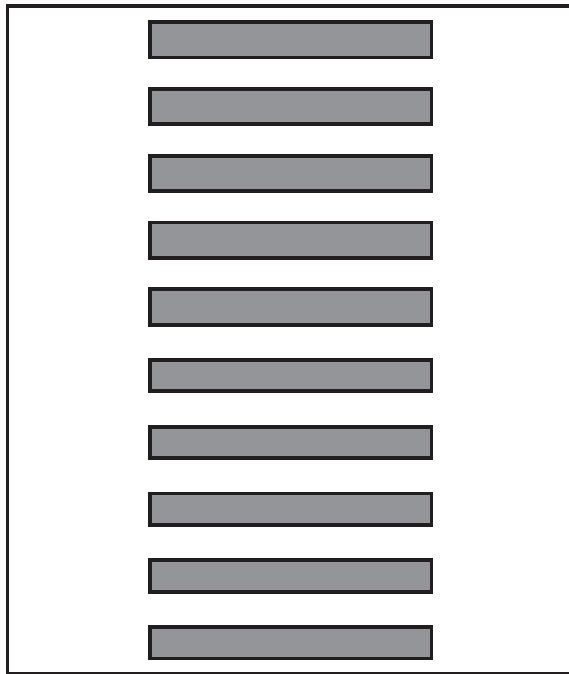
```
// Pernas do Zoog  
stroke(0);
```

```
for (int x = 50; x < 150; x = x + espaco)  
{  
    line(x,y,x,y+altura);  
}
```

Laços de Repetição

Exercício

- Desenhe a figura abaixo



Porém:

- Primeiro retângulo começa em $y = 10$
- Próximo retângulo a uma distância de 10
- Largura = 100 e Altura = 10
- Desenhar retângulos até o fundo da janela

Exercício 1

- Ficou assim?

```
int x = 50, y = 10, largura = 100, altura = 10;  
int espaco = 10;
```

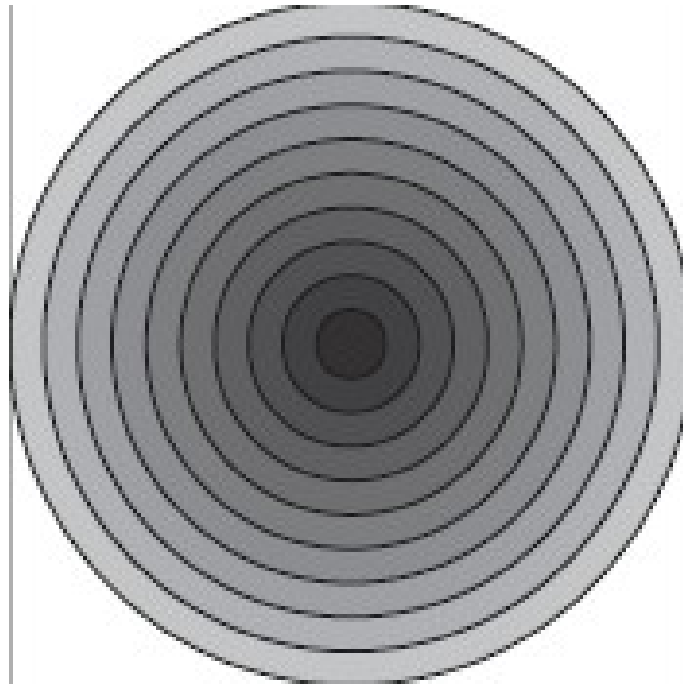
```
void setup()  
{  
  size(200,200);  
  background(255);  
  stroke(0);  
  fill(0);  
}
```

```
void draw ()  
{  
  while (y < height)  
  {  
    rect(x,y,largura,altura);  
    y = y + altura + espaco;  
  }  
}
```

Laços de Repetição

Exercício

Utilize For e While para fazer o desenho abaixo:



Exercício 2

- Ficou assim?

```
int diam, espaco = 10;
```

```
void setup()
```

```
{
```

```
  size(200,200);
```

```
  background(255);
```

```
  stroke(0);
```

```
  fill(175);
```

```
  diam = width;
```

```
}
```

```
void draw ()
```

```
{
```

```
  while (diam > 10)
```

```
  {
```

```
    ellipse(width/2,height/2,diam,diam);
```

```
    diam = diam - 2*espaco;
```

```
  }
```

```
}
```

Exercício 3

- Faça os quadrados abaixo usando **for** e preenchimento aleatório (random)

