

Variáveis, animação e condicionais

...

Programação Criativa - Aula 3
05/10/2015

Conteúdo

- Variáveis
- Principais tipos no Processing
- `setup()` e `draw()`
- Condicionais

**O que é uma
variável?**

O que é uma
variável?



O que é uma variável?

Uma variável é um **objeto** que possui um **nome** e um **tipo**. Ela é utilizada para armazenar um **valor** específico.

Alguns exemplos...

Tipo	Nome	Valor
Número	idade	25
Texto	nome	"Guignard"
Número	posX	520
Número	largura	10.5
Número	pi	3.14

Para declarar (definir/inicializar) uma variável:

tipo nome = valor;

Principais tipos no Processing

Tipo	Exemplo de uso	Descrição
char	<code>char letra= 'a' ;</code>	Uma letra ou um símbolo unicode, como 'a' ou \$. Note que é utilizado aspas simples.
int	<code>int idade = 25;</code>	Um número inteiro. Pode ser positivo ou negativo.
float	<code>float pi = 3.14;</code>	Um número real ou de "ponto flutuante".
boolean	<code>boolean frio = false;</code>	Um valor verdadeiro (true) ou falso (false). Utilizado para operações lógicas.
String	<code>String nome = "Guignard"</code>	Texto. Basicamente uma lista de chars. Note que o S de String é maiúsculo.
color	<code>color c = #FF0000;</code> <code>color t1 = 44;</code> <code>color t2 = color(0,255,0,100);</code>	Cor. Pode ser armazenado dessas três maneiras: hexadecimal, tons de cinza ou definido os valores RGBA utilizando o comando color().

Exemplo

Declarando novas variáveis.

```
size(600,400);  
background(244);  
stroke(44);
```

```
int x = 300;  
int y = 200;  
int d = 200;
```

```
ellipse(x, y, d, d);
```

Exemplo

Atribuindo novos valores à
variáveis já existentes.

Quais os novos valores de d e x?

```
size(600,600);  
background(244);  
stroke(44);  
fill(44, 100);
```

// Inicialização de variáveis

```
int x = 300;
```

```
int y = 300;
```

```
int d = 200;
```

```
ellipse(x, y, d, d);
```

// Atribuição de novos valores

```
d = 100; // lê-se: "x recebe 100"
```

```
x = x - d;
```

```
ellipse(x, y, d, d);
```

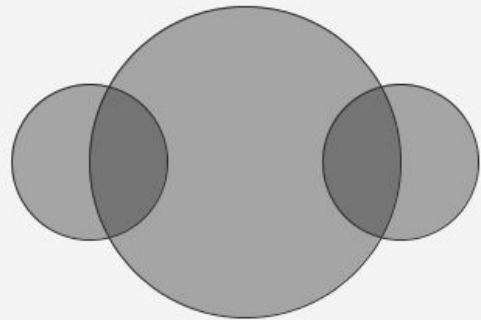
```
 $\overline{x} = x + d * 2;$ 
```

```
ellipse(x, y, d, d);
```

Exemplo

Atribuindo novos valores à
variáveis já existentes.

Quais os novos valores de d e x?



Para "imprimir" o valor de uma variável, use:

println(variavel); // println = print line

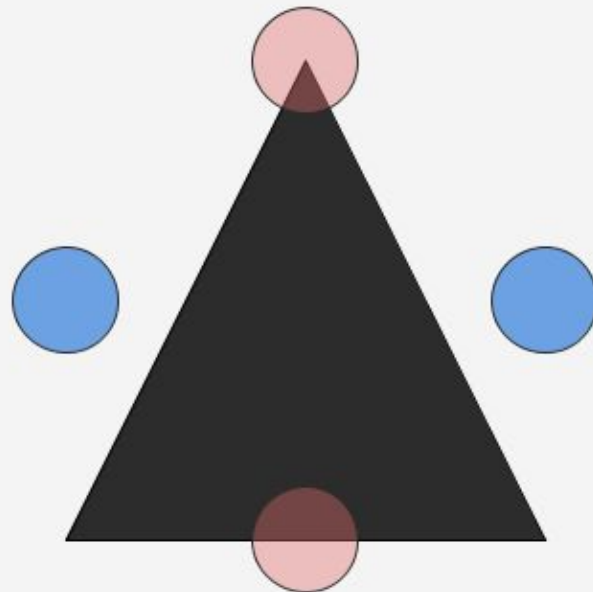
Exemplo:

```
int a = 40;  
println(a);
```

Exercício 1

Desenhe a imagem ao lado utilizando variáveis para todos os valores que utilizar (cor, posição, tamanho do círculo, etc...).

Dica: tente usar o menor número de variáveis possível.



Variáveis de sistema

Tipo	Nome	Conteúdo
int	width	Largura em pixels do canvas
int	height	Altura em pixels do canvas

Exemplo

Usando variáveis do sistema.

Qual será o resultado visual do código ao lado?

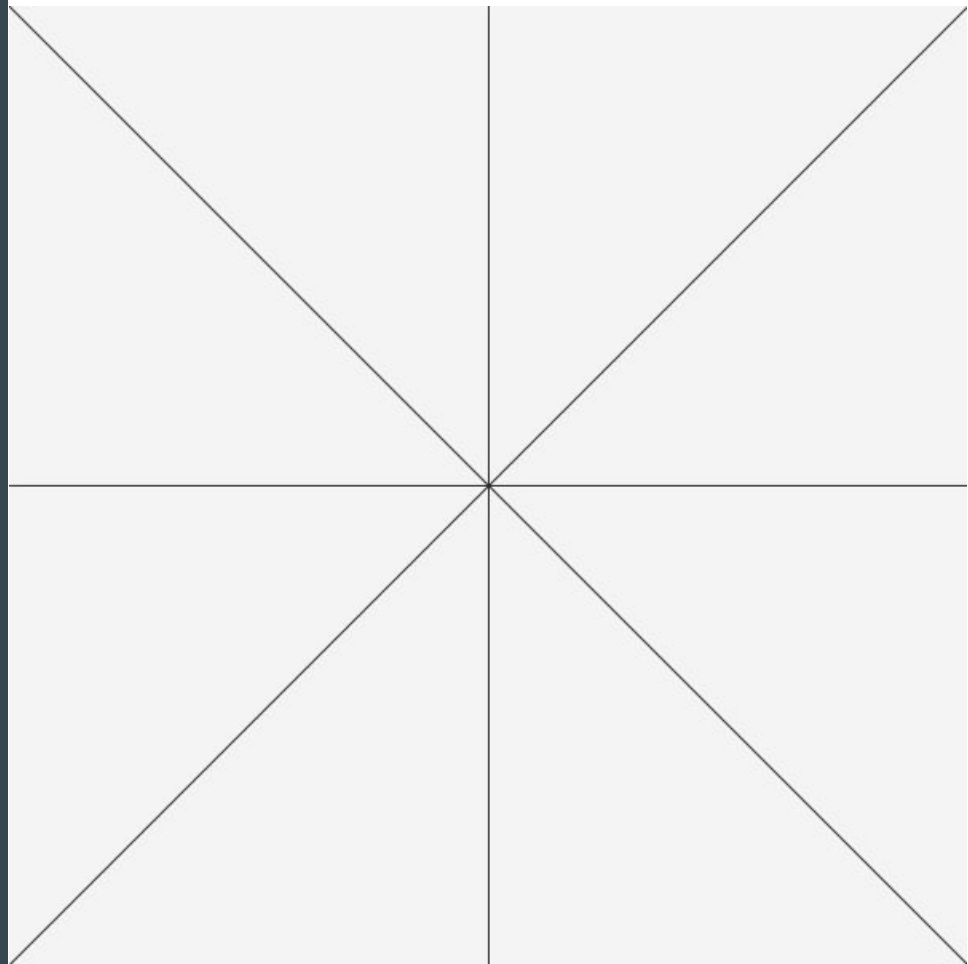
```
size(600,600);  
background(244);  
stroke(44);
```

```
float m = width/2;
```

```
line(m, 0, m, height);  
line(0, m, width, m);  
line(0, 0, width, height);  
line(0, height, width, 0);
```

Exemplo

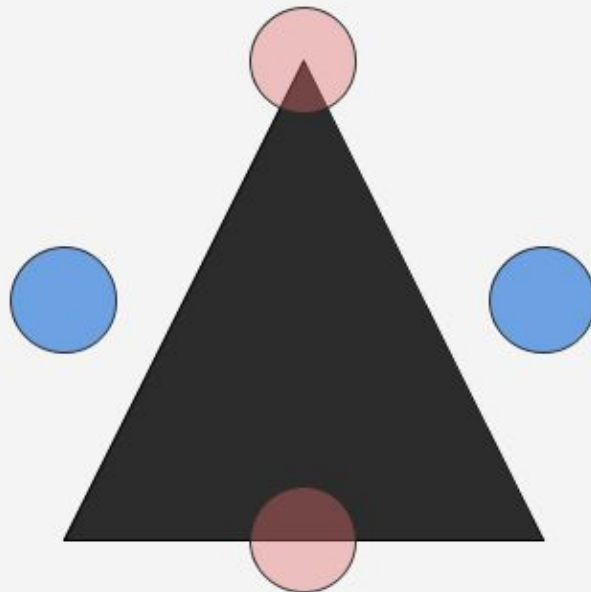
Usando variáveis do sistema.



Exercício 2

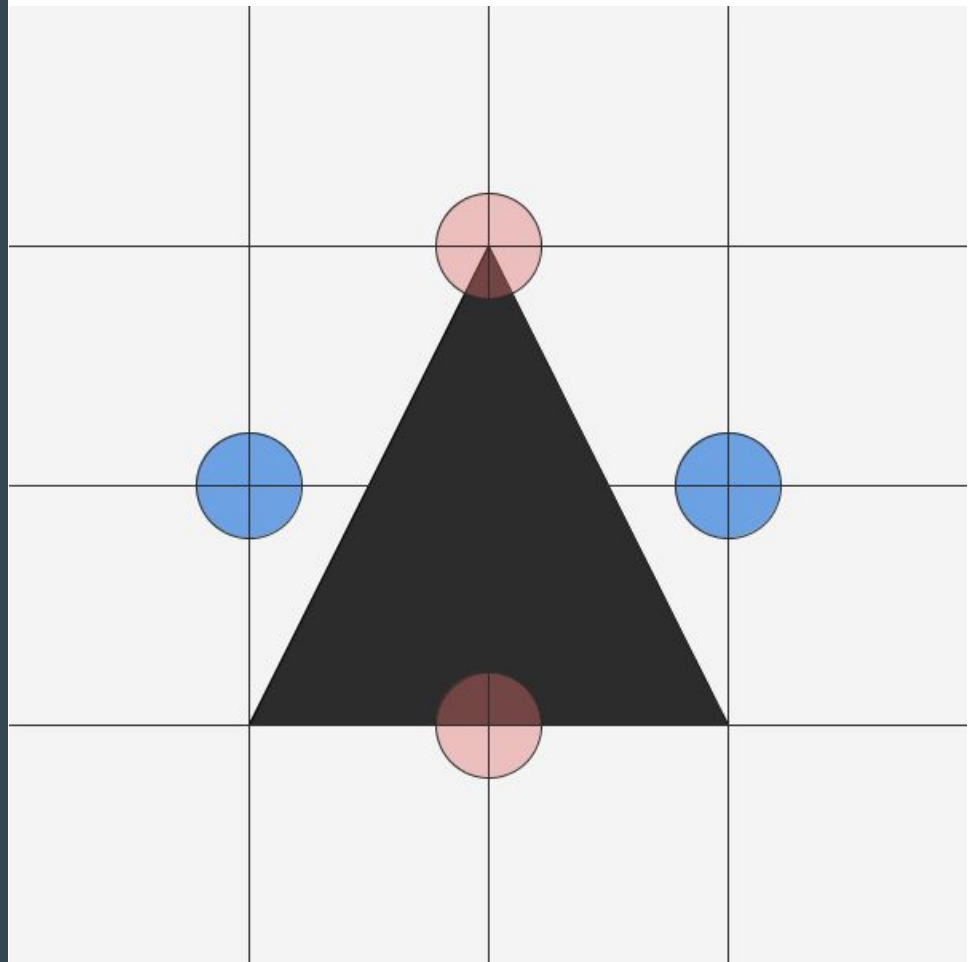
Desenhe a imagem ao lado utilizando variáveis de sistema para todos os valores que utilizar.

Redimensione a imagem para 600x200 para testar.

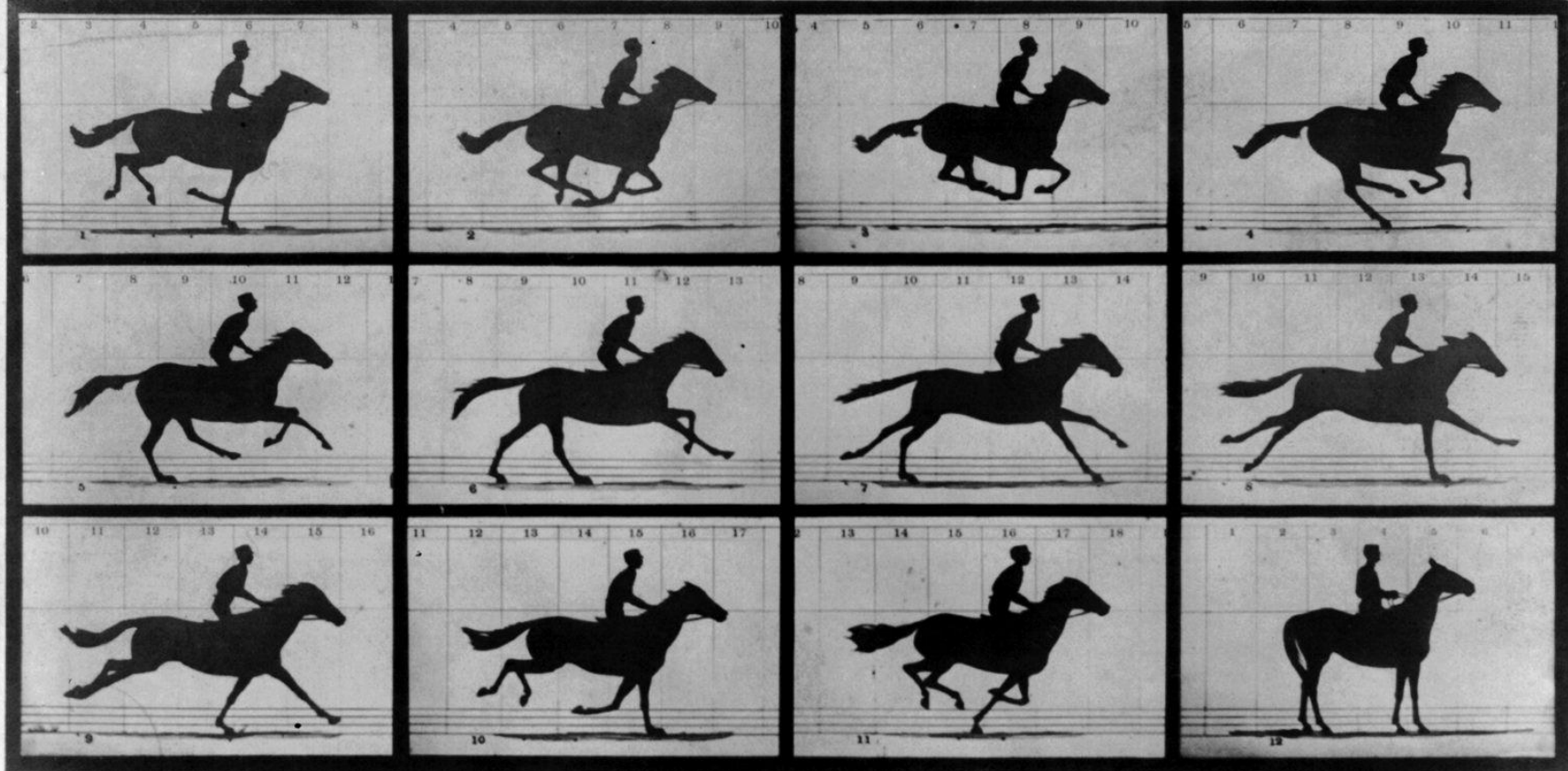


Exercício 2 - Dica

Pense num grid e o que significa você **dividir** o canvas em espaços iguais? Como definir isso em números?



Animacão!



Copyright, 1878, by MUYBRIDGE.

MORSE'S Gallery, 417 Montgomery St., San Francisco.

THE HORSE IN MOTION.

Illustrated by
MUYBRIDGE.

AUTOMATIC ELECTRO-PHOTOGRAPH.

"SALLIE GARDNER," owned by LELAND STANFORD; running at a 1.40 gait over the Palo Alto track, 19th June, 1878.

The negatives of these photographs were made at intervals of twenty-seven inches of distance, and about the twenty-fifth part of a second of time; they illustrate consecutive positions assumed in each twenty-seven inches of progress during a single stride of the mare. The vertical lines were twenty-seven inches apart; the horizontal lines represent elevations of four inches each. The exposure of each negative was less than the two-thousandth part of a second.

Frame loop
setup() & draw()

Bloco de função

```
void setup() {
```

```
}
```

```
void draw() {
```

```
}
```

Bloco de função

// Defina suas variáveis antes do bloco setup()

int x = 200;

// setup() é executado apenas uma vez, no início do programa

void **setup**() {

 size(600,600); // size() deve sempre estar dentro do setup()

 background(244);

}

// draw() é executado uma vez a cada frame (padrão é 60fps [frames por segundo])

void **draw**() {

 // seu código vai dentro deste bloco

}

Variáveis de sistema

Tipo	Nome	Conteúdo
int	width	Largura em pixels do canvas
int	height	Altura em pixels do canvas
int	mouseX	Coordenada horizontal do mouse
int	mouseY	Coordenada vertical do mouse
int	frameCount	Quantos <i>frames</i> já foram desenhados
boolean	mousePressed	Se o mouse está pressionado ou não

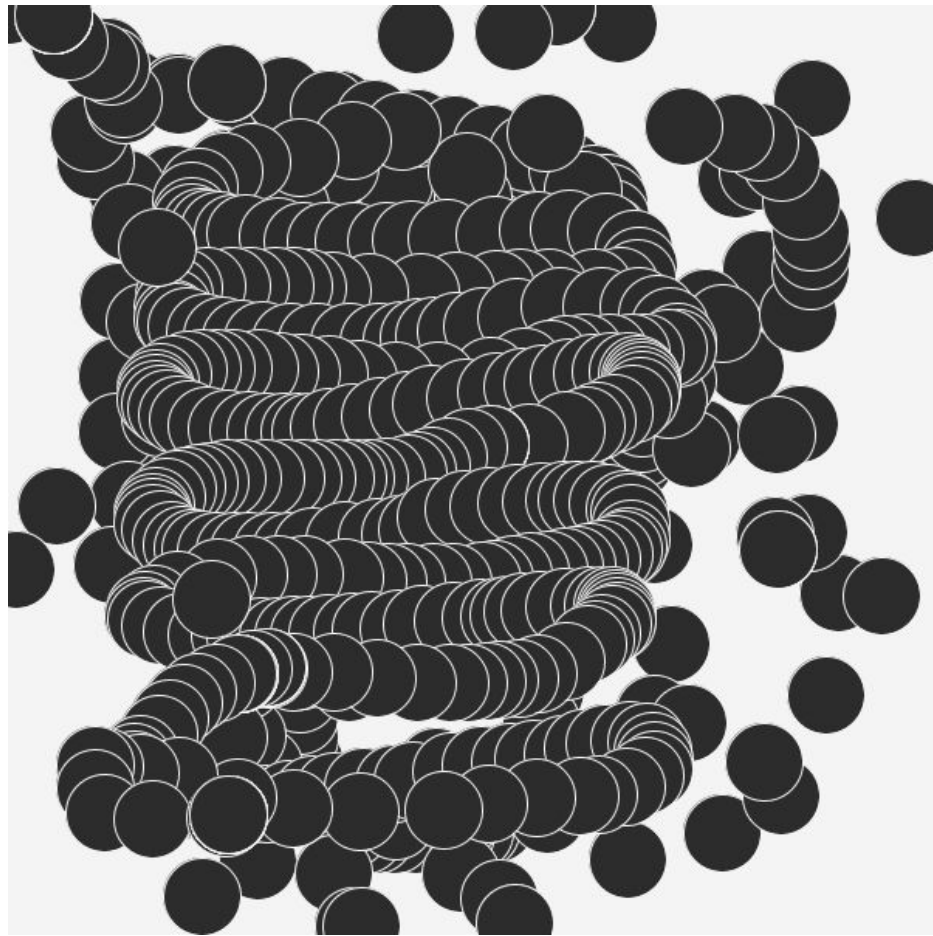
Exemplo

Copie o código ao lado e teste como a função draw() funciona.

```
void setup(){  
    size(600,600);  
    background(244);  
}  
void draw(){  
    stroke(244);  
    fill(44);  
    ellipse(mouseX, mouseY, 50, 50);  
}
```

Exemplo

Copie o código ao lado e teste como a função `draw()` funciona.



Bônus => Altere o sketch anterior e desenhe um retângulo que ocupe a tela inteira com uma opacidade muito baixa (dentro do draw()).

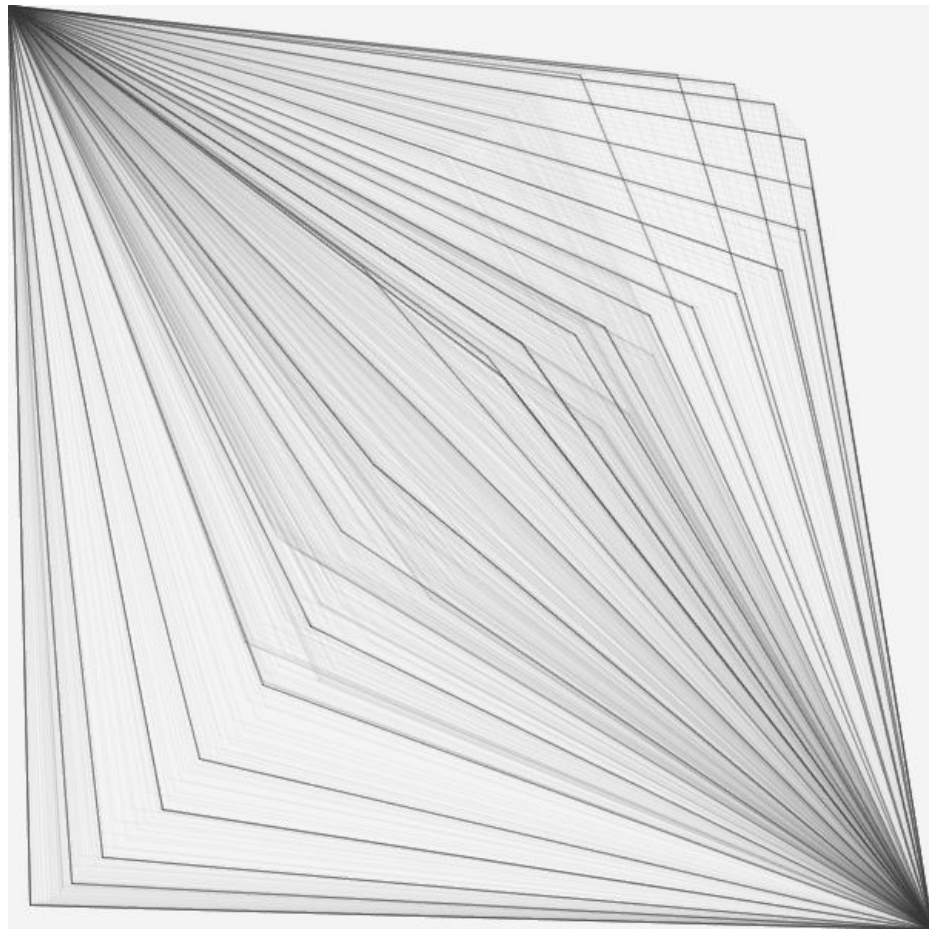
Exercício 3

Desenhe duas linhas que possuem um ponto fixo e outro ponto que segue o mouse.



Exercício 3

Desenhe duas linhas que possuem um ponto fixo e outro ponto que segue o mouse.



Aleatoriedade

Como gerar um valor aleatório
com a função random();

// Gera um float aleatório de 0 à 5

```
float r = random(5);
```

// Gera um float aleatório de 10 à 50

```
random(10, 50);
```

--

// Para converter para int:

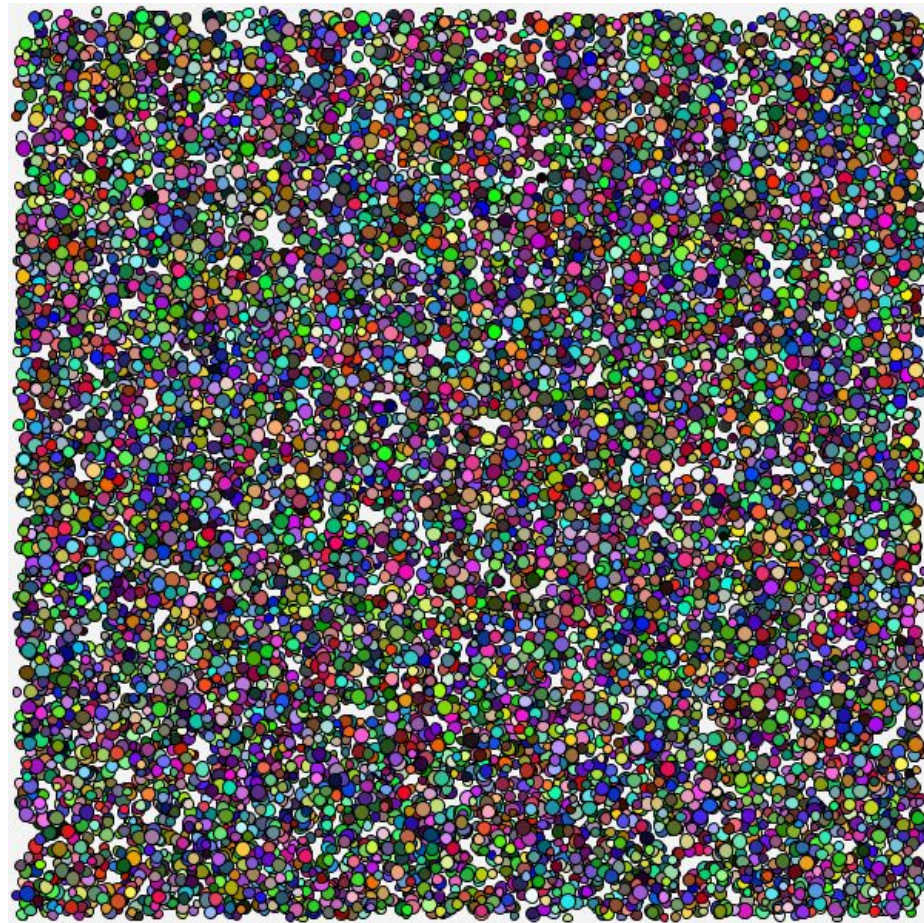
```
int(valorFloat);
```

// Gera um int aleatório de 5 à 10

```
int(random(5, 10));
```

Exercício 3.1

Desenhe círculos em posições aleatórias e com cores aleatórias na função `draw()`.



Exercício 4

Desenhe um círculo no centro da tela e atribua um novo valor pra posição X e Y dele na função draw().

Exemplo: `variavel = variavel + 2;`

Exemplo:

```
int x = 10;
```

...

```
void draw(){
```

```
    ellipse(x, 50, 10, 10);
```

```
    x = x + 1;
```

```
}
```

Condicionais

Expressões booleanas

Uma *expressão booleana* é uma expressão que é verdadeira (true) ou falsa (false):

- Processing é muito chato => false
- Nunca vou aprender a programar => false
- 3 é igual a 3 => true
- 5 é maior que 10 => false
- 50 é menor ou igual a 100 => true

Operadores relacionais

>	maior que	<=	menor ou igual que
<	menor que	==	exatamente igual
>=	maior ou igual que	!=	diferente

Exemplos de expressões booleanas

- $10 > 5$ false
- $3 < 5$ true
- $3 == 3$ true
- $10 == 4$ false
- $10 >= 10$ true
- $1 != 5$ true

Condicional

```
if (expressão_booleana){
```

```
    // código a ser executado caso a expressão seja verdadeira (true)
```

```
}
```

```
else {
```

```
    // código a ser executado caso a expressão seja falsa (false)
```

```
}
```

Exemplo de Condicional

```
int x = 5;
```

```
int y = 10;
```

```
if (x>y){
```

```
    println("O número é maior!");
```

```
}
```

```
else {
```

```
    println("O número é menor!");
```

```
}
```

Exercício 5

Caso o mouse esteja do lado esquerdo da tela, colorir o fundo de preto. Senão colorir o fundo de branco.

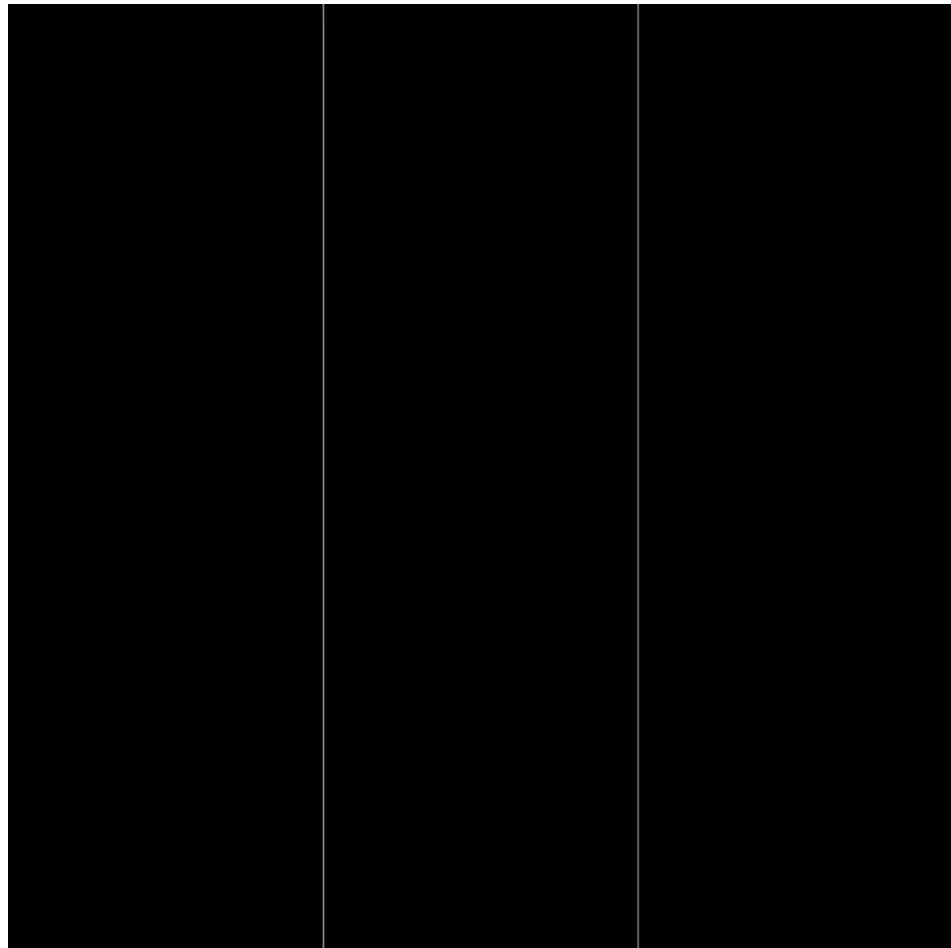
Condicional

```
if (expressão_booleana #1){  
    // código a ser executado caso a expressão #1 seja verdadeira (true)  
}  
else if (expressão_booleana #2){  
    // código a ser executado caso a expressão #2 seja verdadeira (true)  
}  
else if (expressão_booleana #3){  
    // código a ser executado caso a expressão #3 seja verdadeira (true)  
}  
else {  
    // código a ser executado caso nenhuma das expressões acima forem verdadeiras  
}
```


Exercício 6

Se o mouse estiver à esquerda da tela, colorir o fundo de branco; se estiver no centro, colorir de cinza; e à direita, colorir de preto.

Dica: divida a largura do sketch em 3 partes iguais.



Operadores lógicos

Se o mouse estiver do lado direito da tela E o mouse estiver na parte inferior da tela, cora o fundo de branco.

```
if (mouseX > width/2) {  
    if (mouseY > height/2) {  
        background(255);  
    }  
}
```

Operadores lógicos

E lógico	&&
OU lógico	
NÃO lógico	!

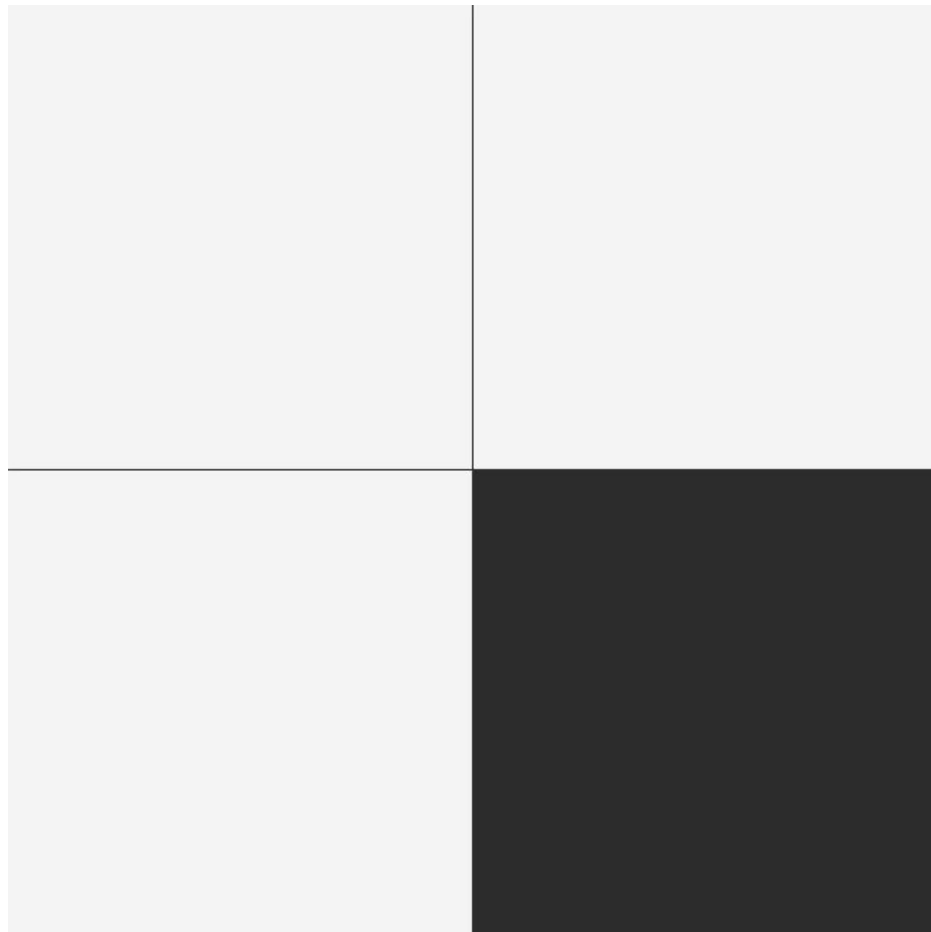
Operadores lógicos

Se o mouse estiver do lado direito da tela E o mouse estiver na parte inferior da tela, cora o fundo de branco.

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

Exercício 7

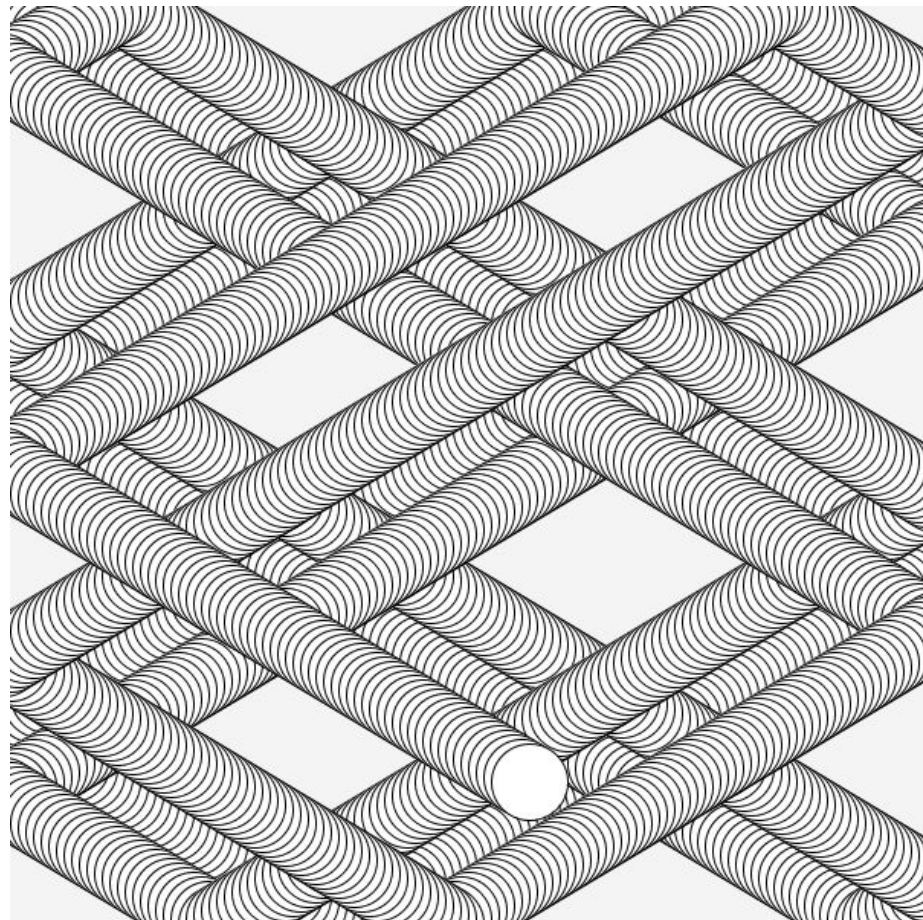
Implemente um "mouse over", ou seja, quando o mouse estiver sobre um dos 4 quadrantes seu fundo deve mudar.



Desafio

Desenhe uma esfera que "quique" quando ela bater na parede.

Dica: use variáveis para determinar posição e velocidade nos dois eixos.



Resumão

Referências

- Generative Art - Matt Pearson
- Learning Processing - Daniel Shiffman
- <https://processing.org/reference/>
- pack by Martin Vanko from the Noun Project
-