

Atividade 6 - Processing

1. Proposta

Crie um programa com ao menos quatro fases ou cenários diferentes, organizadas com algum nível de modularização conforme trabalhado em sala (funções em um mesmo arquivo, funções em arquivos separados, classes em arquivos separados usando uma mesma interface). Entregue um único arquivo PDF com um parágrafo explicando como fez a atividade e com prints do código e do resultado obtido.

2. Código

Atividade 6	desenho1	desenho2	desenho3	desenho4	▼
-------------	----------	----------	----------	----------	---

```
Desenho1 desenho1;  
Desenho2 desenho2;  
Desenho3 desenho3;  
Desenho4 desenho4;  
int contador = 0;  
int framestroca = 60;  
  
void setup() {  
    size(800, 600);  
    desenho1 = new Desenho1();  
    desenho2 = new Desenho2();  
    desenho3 = new Desenho3();  
    desenho4 = new Desenho4();  
}
```

- São declarados quatro objetos de classes de desenhos (Desenho1, Desenho2, Desenho3, Desenho4).
- "contador" é usado para contar o número de frames que se passaram.
- "framestroca" define quantos frames cada desenho será exibido (60 frames = 1 segundo).

```

void draw() {
    background(255);
    contador++;

    if (contador / framestroca % 4 == 0) {
        desenho1.display();
    } else if (contador / framestroca % 4 == 1) {
        desenho2.display();
    } else if (contador / framestroca % 4 == 2) {
        desenho3.display();
    } else if (contador / framestroca % 4 == 3) {
        desenho4.display();
    }
}

```

O método draw é chamado continuamente, formando o loop principal do programa, background(255) limpa a tela a cada frame, preenchendo-a com a cor branca, frameCounter++ incrementa o contador de frames a cada iteração.

A expressão frameCounter / changeInterval % 4 calcula qual desenho deve ser exibido com base no número de frames que se passaram, dividindo o contador pelo intervalo de mudança e pegando o resto da divisão por 4:

- Se o resto for 0, exibe o desenho1.
- Se o resto for 1, exibe o desenho2.
- Se o resto for 2, exibe o desenho3.
- Se o resto for 3, exibe o desenho4.

Atividade 6	desenho1	desenho2	desenho3	desenho4	▼
<pre> class Desenho4 { void display() { fill(255, 255, 0); line(0, 0, width, height); line(0, height, width, 0); } } </pre>					

A classe Desenho1 tem um método display que desenha um círculo vermelho no centro da tela.

Atividade 6	desenho1	desenho2	desenho3	desenho4	▼
-------------	----------	----------	----------	----------	---

```

class Desenho3 {
    void display() {
        fill(0, 0, 255);
        triangle(width / 2, height / 2 - 50, width / 2 - 50, height / 2
    }
}

```

A classe Desenho2 tem um método display que desenha um quadrado verde no centro da tela.

Atividade 6	desenho1	desenho2	desenho3	desenho4	▼
-------------	----------	----------	----------	----------	---

```

class Desenho2 {
    void display() {
        fill(0, 255, 0);
        rect(width / 2 - 50, height / 2 - 50, 100, 100);
    }
}

```

A classe Desenho3 tem um método display que desenha um triângulo azul no centro da tela.

Atividade 6	desenho1	desenho2	desenho3	desenho4	▼
-------------	----------	----------	----------	----------	---

```

class Desenho1 {
    void display() {
        fill(255, 0, 0);
        ellipse(width / 2, height / 2, 100, 100);
    }
}

```

A classe Desenho4 tem um método display que desenha duas linhas amarelas cruzando a tela.

3. Resultado

