

Atividade 5 - Processing

1. Proposta

Adapte o exemplo de uso de classe construído em sala de aula (coleção de ellipses com tamanho, posições e cores aleatórias) para que os objetos fiquem se movimentando aleatoriamente na tela. Entregue um único arquivo PDF com um parágrafo explicando como fez a atividade e com prints do código e do resultado obtido.

2. Código

A classe `Ellipse` define os atributos e comportamentos das elipses que serão desenhadas na tela. Cada objeto da classe `Ellipse` tem os seguintes atributos:

- `px` e `py`: coordenadas de posição da ellipse.
- `cor`: cor da ellipse.
- `tamanho`: tamanho (diâmetro) da ellipse.
- `speedX` e `speedY`: velocidades da ellipse nos eixos X e Y.

A classe `Ellipse` inicializa esses atributos com os valores aleatórios.

```
class Ellipse {
    float px;
    float py;
    color cor;
    float tamanho;
    float speedX;
    float speedY;

    Ellipse(float x, float y, color c, float t, float sx, float sy){
        px = x;
        py = y;
        cor = c;
        tamanho = t;
        speedX = sx;
        speedY = sy;
    }
}

int qtd;
Ellipse ellipses[];
```

As variáveis globais são:

- qtd: número de elipses a serem desenhadas.
- elipses[]: array de objetos Ellipse que armazenará todas as elipses.

A função setup é executada uma vez no início do programa. Aqui, a tela é configurada com tamanho 800x600 e sem bordas. A variável qtd é definida como 40, indicando que serão criadas 40 elipses. O array elipses é inicializado com o tamanho definido por qtd.

Dentro de um laço for, cada ellipse é criada com valores aleatórios para posição (px e py), tamanho (tamanho), cor (cor) e velocidades (speedX e speedY). Esses objetos Ellipse são armazenados no array elipses.

```
void setup(){
  size(800,600);
  noStroke();
  qtd = 40;
  elipses = new Ellipse[qtd];
  for(int i = 0; i < qtd; i++){
    float px = random(50, 750);
    float py = random(50, 550);
    float tamanho = random(10, 100);
    color cor = color(random(0,255), random(0,255), random(0,255), random(0,255));
    float speedX = random(-1, 1);
    float speedY = random(-1, 1);
    elipses[i] = new Ellipse(px, py, cor, tamanho, speedX, speedY);
  }
}
```

A função draw é executada continuamente para redesenhar a tela e criar a animação.

fill(elipses[i].cor) define a cor de preenchimento.

ellipse(elipses[i].px, elipses[i].py, elipses[i].tamanho, elipses[i].tamanho) desenha a ellipse na posição e tamanho especificados.

Atualização da Posição: As posições px e py são atualizadas com base nas velocidades speedX e speedY.

Se a ellipse tocar as bordas da tela (esquerda, direita, cima ou baixo), a velocidade correspondente é invertida.

A verificação é feita checando se px - tamanho é menor que 0 ou px + tamanho é maior que a largura da tela, e se py - tamanho é menor que 0 ou py + tamanho é maior que a altura da tela.

Se uma colisão for detectada, a cor da ellipse é alterada para uma nova cor aleatória.

```

void draw(){
    background(255);

    for(int i = 0; i < qtd; i++){
        fill(ellipses[i].cor);
        ellipse(ellipses[i].px, ellipses[i].py, ellipses[i].tamanho, ellipses[i].tamanho);

        ellipses[i].px += ellipses[i].speedX;
        ellipses[i].py += ellipses[i].speedY;

        boolean tocouBorda = false;

        if (ellipses[i].px - ellipses[i].tamanho < 0 || ellipses[i].px + ellipses[i].tamanho > width) {
            ellipses[i].speedX *= -1;
            tocouBorda = true;
        }
        if (ellipses[i].py - ellipses[i].tamanho < 0 || ellipses[i].py + ellipses[i].tamanho > height) {
            ellipses[i].speedY *= -1;
            tocouBorda = true;
        }

        if (tocouBorda) {
            ellipses[i].cor = color(random(0,255), random(0,255), random(0,255), random(0,255));
        }
    }
}

```

3. Resultado

Dessa forma, o programa cria uma animação onde elipses se movem pela tela, mudando de cor ao colidirem com as bordas.

