

Projeto – Tempestade sobre a Normandia

RESUMO DA IDEIA

Uma simulação em Processing onde um pelotão de paraquedistas salta durante uma tempestade, com vento, obstáculos, colisões, interação com o utilizador e regras físicas criativas.

ELEMENTOS PRINCIPAIS

- Fundo animado com nuvens usando noise()
- Mar com animação de ondas
- Paraquedistas com massa, velocidade inicial, resistência do ar e paraquedas
- Obstáculos como pássaros ou bombas
- Mundo com paredes (choque elástico)
- Interação: vento com o rato, abrir/fechar paraquedas, teclas para alterar gravidade ou tempestade

PSEUDOCÓDIGO FORMATADO

CLASSE Paraquedista

```
Atributos:  
x, y           // posição  
vx, vy        // velocidade  
ax, ay        // aceleração  
massa  
tamanho_paraquedas  
paraquedas_aberto // boolean  
vivo          // boolean  
cor  
  
Métodos:  
aplicarForca(fx, fy)  
atualizar()  
desenhar()
```

LISTAS PRINCIPAIS

PARAQUEDISTAS
OBSTACULOS

FUNÇÃO setup()

```
definir tamanho da janela  
  
criar N paraquedistas:  
    para i de 1 até N:  
        p = novo Paraquedista
```

```

p.x = valor_aleatorio_no_topo
p.y = -20
p.vx = random(-1, 1)
p.vy = random(0, 2)
p.massa = random(0.8, 2)
p.tamanho_paraquedas = random(20, 40)
p.paraquedas_aberto = true
adicionar p a PARAQUEDISTAS

gerar obstáculos e adicionar a OBSTACULOS

```

FUNÇÃO draw()

```

desenhar fundo animado (nuvens, noise)
desenhar mar

para cada obstáculo O em OBSTACULOS:
    O.atualizar()
    O.desenhar()

para cada paraquedista P em PARAQUEDISTAS:

    gravidade = 0.1 * P.massa
    P.aplicarForca(0, gravidade)

    se P.paraquedas_aberto:
        resistencia = -0.05 * P.vy
    senão:
        resistencia = -0.01 * P.vy
    P.aplicarForca(0, resistencia)

    vento = map(noise(P.y * 0.01, frameCount * 0.01), 0..1, -0.2..0.2)
    P.aplicarForca(vento, 0)

    para cada obstáculo O:
        se P colide com O:
            aplicar efeitos

    se P.x < 0 ou P.x > largura:
        P.vx = P.vx * -0.9

    se P.y >= altura - nivel_do_mar:
        P.vivo = false
        criar splash
        continuar

    P.atualizar()
    P.desenhar()

```

FUNÇÃO mouseDragged()

```

para cada P:
    se distancia(mouse, P) < 80:
        forçaX = (mouseX - pmouseX) * 0.2
        forçaY = (mouseY - pmouseY) * 0.2
        P.aplicarForca(forçaX, forçaY)

```

FUNÇÃO mousePressed()

```

para cada P:
    se mouse próximo de P:
        P.paraquedas_aberto = não P.paraquedas_aberto

```

FUNÇÃO keyPressed()

```
se tecla == 'g': alterar gravidade  
se tecla == 't': ativar tempestade  
se tecla == 'r': reiniciar simulação
```

MÉTODOS DA CLASSE Paraquedista

```
aplicarForca(fx, fy):  
    ax += fx / massa  
    ay += fy / massa  
  
atualizar():  
    vx += ax  
    vy += ay  
    x += vx  
    y += vy  
    ax = 0  
    ay = 0  
  
desenhar():  
    desenhar corpo  
    se paraquedas_aberto: desenhar paraquedas
```