

Universidade Católica Portuguesa

13 de Novembro de 2007

PROGRAMAÇÃO MULTIMÉDIA

Docente: Jorge Cardoso

Curso de Som e Imagem

(Tempo permitido: 2h00m)

NOTA: A realização deste teste pode ser feita com recurso aos apontamentos e à referência do Processing.

Em todos os exercícios deve ser utilizada a sintaxe do Processing.

Nome: _____

1. (20%) Construa os ciclos (incluindo declaração de variáveis) necessários para percorrer e mostrar na totalidade as seguintes estruturas de dados:

(a) `byte idades [];`
`idades = new byte [30];`

```
for (int i = 0; i < 30; i++) {  
    println(idades[i]);  
}
```

(b) `float pesos [][];`
`pesos = new float [95] [10];`

```
for (int i = 0; i < 95; i++) {  
    for (int j = 0; j < 10; j++) {  
        println(pesos[i][j]);  
    }  
}
```

2. (20%) Construa um ciclo (incluindo declaração de variáveis) que some 5 a todos os elementos da seguinte matriz (o resultado deve ser colocado noutra matriz):

```
float pesos[][];  
pesos = new float[50][40];
```

```
float resultado[][];  
resultado = new float[50][40];  
  
for (int i = 0; i < 50; i++) {  
    for (int j = 0; j < 40; j++) {  
        resultado[i][j] = pesos[i][j] + 5;  
    }  
}
```

3. (20%) Altere o ciclo anterior de modo a que apenas seja adicionado 5 às posições que contêm um valor negativo. Os valores positivos devem ser mantidos iguais ao da matriz **pesos**.

```
float resultado[][];  
resultado = new float[50][40];  
  
for (int i = 0; i < 50; i++) {  
    for (int j = 0; j < 40; j++) {  
        if (pesos[i][j] < 0) {  
            resultado[i][j] = pesos[i][j] + 5;  
        } else {  
            resultado[i][j] = pesos[i][j];  
        }  
    }  
}
```

4. (10%) O seguinte programa em Processing contém erros sintácticos. Identifique-os, corrigindo as linhas em que se encontram:

```
1 void setup() {
2   size(500, 500);
3   frame Rate(10);
4 }
5
6
7 voiddraw() {
8   for (inti = 0; i < 100; i++) {
9     ellipse(random(500), random(500), random(10), random(10));
10  }
11 }
```

```
1 void setup() {
2   size(500, 500); // '500' e não '500'
3   frameRate(10); // 'frameRate' e não 'frame Rate'
4 }
5
6
7 void draw() { // 'void draw' e não 'voiddraw'
8   for (int i = 0; i < 100; i++) { // 'int i' e não 'inti'
9     ellipse(random(500), random(500), random(10), random(10)); // '
10     random(500)' e não 'random(500)'
11 }
```

5. (5%) Explique o que faz o programa anterior:

O programa desenha, em cada frame, 100 elipses em posições aleatórias dentro da janela. As dimensões da elipse também são aleatórias entre 0 e 10 píxeis. O conteúdo da janela não é apagado, pelo que as elipses das frames anteriores estão sempre presentes.

6. (20%) Altere o programa de modo a que:

- (a) As elipses sejam desenhados com cores aleatórias
- (b) Sejam apenas desenhados 5 elipses de cada vez que `draw()` executa
- (c) O tamanho máximo (largura e altura) possível de cada elipse seja 50 pixels.

```
void setup() {  
  size(500, 500);  
  frameRate(10);  
}  
  
void draw() {  
  for (int i = 0; i < 5; i++) {  
    fill(random(255), random(255), random(255));  
    ellipse(random(500), random(500), random(50), random(50));  
  }  
}
```

7. (5%) Escreva em Processing um programa que:

- (a) Possua uma janela gráfica com dimensão **500x500** píxels.
- (b) Anime um círculo dentro do ecrã com movimento em X e em Y.
- (c) A velocidade em X do círculo deve aumentar da esquerda para a direita e diminuir da direita para a esquerda.
- (d) A velocidade em Y do círculo deve diminuir de cima para baixo e aumentar de baixo para cima.

O nome do projecto Processing deverá ser TestePM<Primeiro><Ultimo> (em que <Primeiro> e <Ultimo> são os seus primeiro e último nomes, respectivamente). O projecto deverá ser gravado no *desktop* do seu computador.

```
float x, y;      // posição do círculo
float vx, vy;   // velocidade

boolean sx, sy; // indica a direcção do movimento: true - direita/baixo

void setup() {
  size(500, 500);
  frameRate(10);

  x = 100;
  y = 100;
  vx = 1;
  vy = 1;
  sx = true;
  sy = true;
}

void draw() {
  ellipse(x, y, 30, 30);

  x = x + vx;
  y = y + vy;

  // se bateu invertemos a direcção
  if (x > 500 || x < 0) {
    sx = !sx;
  }
  if (y > 500 || y < 0) {
    sy = !sy;
  }

  // consoante a direcção, a velocidade é positiva
  // ou negativa e proporcional à posição.
  // Somamos dois para garantir que nunca é zero (o círculo
  // parava)
  if (sx) {
    vx = x/100+2;
  } else {
    vx = -x/100-2;
  }
  if (sy) {
    vy = y/100+1;
  } else {
    vy = -y/100-1;
  }
}
```
