



Escola Superior de Tecnologia e Gestão
Instituto Politécnico da Guarda

ENUNCIADO DE AVALIAÇÃO

Modelo
PED.002.02

Curso	Engenharia Informática				Ano letivo	2019/2020	
Unidade curricular	Computação Gráfica						
Ano curricular	3º	Semestre	1º S	Data	14/02/2020	Duração	2h30

Exame Normal

Este exame tem um peso de 60% na nota final.

1. Escreva uma função em Java para desenhar a curva dada pelas equações paramétricas indicadas ao lado.

$$x = x_0 + a \cos t$$

$$y = y_0 + b \sin t$$

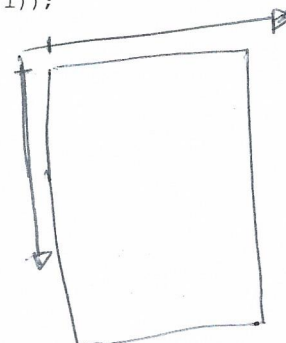
$$\min < t < \max$$

2. Desenhe um esboço que represente o resultado do seguinte código. Indique no esboço as coordenadas de todos os pontos de controle.

```
public void paintComponent(Graphics g) {  
    super.paintComponent(g);  
    Graphics2D g2 = (Graphics2D) g;  
    g2.translate(getWidth()/2, getHeight()/2);  
    GeneralPath p = new GeneralPath(GeneralPath.WIND_EVEN_ODD);  
    p.moveTo(-50f, -50f);  
    p.quadTo(-85f, 0f, -50f, 50f);  
    p.lineTo(50, 50);  
    p.curveTo(10, 25, 90, -25, 50, -50);  
    p.lineTo(-50, -50);  
    p.closePath();  
    GradientPaint paint = new GradientPaint(-50, -50, Color.BLACK, 0, 0, Color.WHITE,  
    true);  
    g2.setPaint(paint);  
    g2.fill(p);  
}
```

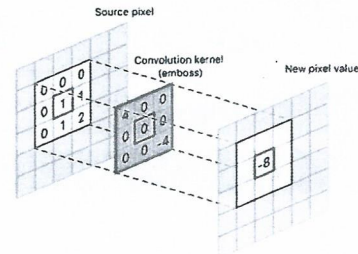
3. Desenhe um esboço que represente o resultado do seguinte código. Indique no esboço os eixos do sistema de coordenadas.

```
public void paintComponent(Graphics g) {  
    super.paintComponent(g);  
    Graphics2D g2 = (Graphics2D) g;  
    int w = this.getWidth();  
    int h = this.getHeight();  
    int l = 100;  
    BufferedImage image = getImage("test2d/images/Smile.jpg");  
    Paint paint = new TexturePaint(image, new Rectangle2D.Double(-2*l, -l, 2*l,  
    2*l));  
    Area a = new Area(new Rectangle2D.Double(-l, -l, 2*l, 2*l));  
    Area b = new Area(new Rectangle2D.Double(-l/2, -l/2, l, l));  
  
    AffineTransform at = new AffineTransform();  
    at.rotate(Math.toRadians(45));  
    b=b.createTransformedArea(at);  
    a.subtract(b);  
  
    g2.translate(w/2, h/2);  
    g2.rotate(Math.toRadians(45));  
  
    g2.setPaint(paint);  
    g2.fill(a);  
    g2.setColor(Color.black);  
    g2.draw(a);  
}
```





4. A operação de convolução aplica um filtro específico a uma imagem. O filtro é representado por uma matriz chamada Kernel (geralmente com tamanho 3x3). Dê um exemplo de um kernel para aplicar um filtro que causa o chamado efeito de suavização (usado para eliminar “grainha” na imagem, por exemplo) e explique sucintamente como é que o filtro funciona e consegue suavizar a imagem.



5. Considere que se pretende aplicar um fator de escala (E_x , E_y) a um objeto 2D qualquer, mas de modo a que as coordenadas do centro do objeto (X_c , Y_c), não se alterem com a transformação. Isto é, pretende-se que o objeto fique centrado no mesmo ponto, após a aplicação da transformação. Apresente a matriz 3x3 da transformação composta que ao ser multiplicada pelos pontos do objeto produz o resultado desejado.
6. Complete o código a seguir para criar uma animação na qual um objeto quadrado roda em torno do centro da janela numa orbida de raio igual a 100.

```
class MyPanel extends JPanel implements Runnable { // 1º
    float ang = 0;

    public MyPanel() {
        setPreferredSize(new Dimension(400, 400)); // 2º
        // 3º
    }

    public void paintComponent(Graphics g) {
        super.paintComponent(g);
        Graphics2D g2 = (Graphics2D) g; // 4º

        g2.drawOval(-100, -100, 200, 200);

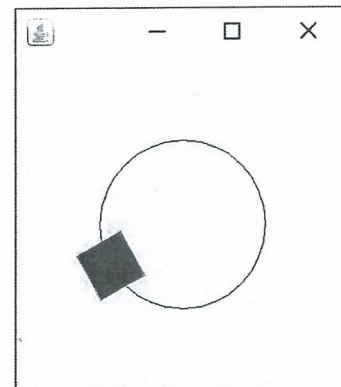
        Shape s = new Rectangle2D.Double(-30, -30, 60, 60);
        AffineTransform tx = new AffineTransform(); // 5º
        tx.set // 6º
        tx. // 7º
        s = // 8º

        g2.setPaint(Color.RED);
        g2.fill(s);

    }

    public void // 9º
        while (true) { // 10º

            try {
                Thread.sleep(50);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
}
```





7. Desenhe a figura correspondente à geometria definida pelo código a seguir. Deve indicar no desenho as coordenadas e os índices dos vértices. Desenhe também as normais da geometria.

```
GeometryInfo gi = new GeometryInfo(GeometryInfo.TRIANGLE_STRIP_ARRAY);
int[] stripIndex = { 4, 4, 4 };
gi.setStripCounts(stripIndex);
Point3f[] giCoords = new Point3f[7];
giCoords[0] = new Point3f(0f, 0f, 0f);
giCoords[1] = new Point3f(0f, 0f, 1f);
giCoords[2] = new Point3f(0f, 1f, 1f);
giCoords[3] = new Point3f(0f, 1f, 0f);
giCoords[4] = new Point3f(1f, 1f, 0f);
giCoords[5] = new Point3f(1f, 0f, 0f);
giCoords[6] = new Point3f(1f, 0f, 1f);
gi.setCoordinates(giCoords);
int[] giIndices = { 0, 3, 1, 2, 0, 5, 3, 4, 1, 6, 0, 5 };
gi.setCoordinateIndices(giIndices);

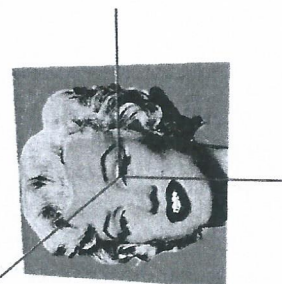
NormalGenerator ng = new NormalGenerator();
ng.generateNormals(gi);
```

Considere que a cena é iluminada por uma luz ambiente de cor branca e o objeto usa a seguinte aparência.

```
Appearance ap = new Appearance();
ap.setMaterial(new Material());
```

8. Complete o código a seguir para criar um painel quadrado, onde é mapeada uma parte da imagem de textura, conforme mostrado na figura ao lado. Considere que todos os vértices do painel têm coordenada z = 0;

```
QuadArray p = new QuadArray(4, QuadArray.COORDINATES); //1°
p.setCoordinate(0, new Point3d(-1, -1, 0));
p.setCoordinate(1, new Point3d(1, 0)); //2°
p.setCoordinate(2, new Point3d(1, 0)); //3°
p.setCoordinate(3, new Point3d(0, 0)); //4°
p.setTextureCoordinate(0, 0, new TexCoord2f(0, 0)); //5°
p.setTextureCoordinate(0, 1, new TexCoord2f(1, 0)); //6°
p.setTextureCoordinate(0, 2, new TexCoord2f(1, 1)); //7°
p.setTextureCoordinate(0, 3, new TexCoord2f(0, 1)); //8°
```



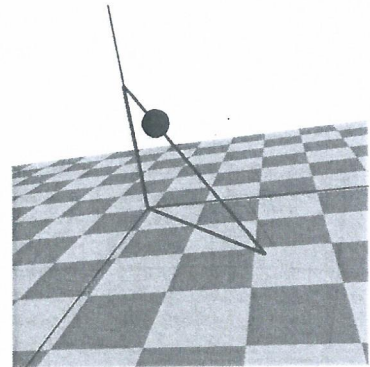
9. Desenhe o grafo de cena de um programa que usa um objeto DistanceLOD para configurar três níveis de detalhes para exibir um objeto de texto 3D da string "Java". O primeiro nível usa uma aparência do tipo Material para iluminar o objeto com luz. O segundo nível usa uma aparência com base numa cor única. O terceiro nível substitui o objeto por uma Box com a mesma aparência usada no segundo nível. Considere também que existe um objeto MouseRotate para interagir com o objeto.
10. Considere que um ponto de coordenadas (1, 0, 0) na superfície de um objeto tem um normal na direção (0, 1, 0) e é iluminado por uma luz com intensidade (1, 1, 0), localizada em (0, 2, 0). A vista está posicionada em (5, 3, 0). Se os coeficientes de reflexão especular do material escolhido para a aparência do objeto são (0,3, 0,5, 0,2) e o índice de brilho é 94, qual é o valor da reflexão especular nesse ponto? Não é necessário calcular os valores, apenas apresentar as equações. Como caracterizaria o material deste objeto? Muito brilhante ou pouco brilhante? Justifique.

$$I = I_a K_a + I_p K_d \cos \theta + I_p K_s \cos^n \alpha$$

$I_a =$



11. Complete o código a seguir para programar um comportamento de PositionPathInterpolator que faz com que uma esfera siga o caminho triangular mostrado na figura ao lado. Pretende-se que a esfera leve 4 segundos para percorrer a hipotenusa do triângulo e 3 segundos para percorrer cada uma das pernas do triângulo. O movimento da esfera será sempre na mesmo sentido? Justifique.



```
...  
Appearance ap = new Appearance(); //1°  
Primitive sphere = new Sphere(0.05f,  
Primitive, 24, ap); //2°  
  
TransformGroup t = new TransformGroup();  
t. //3°  
t.addChild(sphere);  
root.addChild(t);  
  
Alpha a = new Alpha(-1, 10000);  
a.setMode(Alpha.INCREASING_ENABLE |  
Alpha.DECREASING_ENABLE);  
a.setDecreasingAlphaDuration(10000);  
float[] k = //4°  
Point3f[] p = {new Point3f(0f, 1f, 0f), } //5°  
  
tr = //6°  
PositionPathInterpolator i = new  
PositionPathInterpolator(a, t, tr, k, p); //7°  
i. //8°  
t. //9°  
...
```

12. A API Java 3D permite implementar objetos com geometrias que podem ser alteradas dinamicamente durante a execução da aplicação. Isso pode ser feito de um modo mais restritivo através da classe **Morph**, ou de um modo mais genérico e personalizado através da implementação da classe interface **GeometryUpdater** e da criação de uma classe que implementa um **behavior personalizado**. Considere o segundo modo e descreva sucintamente, por palavras suas, como se pode implementar esse segundo modo com base nas duas classes referidas.