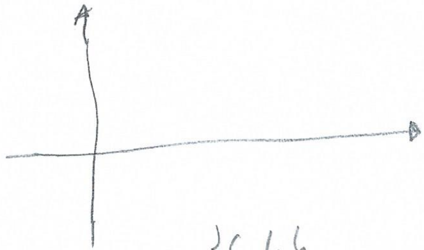
Escola Superior de Tecnologia e Gestão Instituto Politécnico da Guarda	ENUNCIADO DE AVALIAÇÃO	Modelo PED.002.02

Curso	**Engenharia Informática**					*Ano letivo*	2018/2019
Unidade curricular	**Computação Gráfica**						
Ano curricular	3º	*Semestre*	1º S	*Data*	01/02/2019	*Duração*	2h00

Frequência 3D

1. (2) Draw the figure corresponding to the geometry defined by the following code.

```
int[] stripIndexCounts = {6, 4};
IndexedTriangleStripArray geom = new IndexedTriangleStripArray(8,
        GeometryArray.COORDINATES, 10, stripIndexCounts);
        Point3f[] coords = new Point3f[8];
        coords[0] = new Point3f(-1f, 2f, 0f);
        coords[1] = new Point3f(-1f, 1f, 0f);
        coords[2] = new Point3f(0f, 2f, 0f);
        coords[3] = new Point3f(0f, 1f, 0f);
        coords[4] = new Point3f(1f, 2f, 0f);
        coords[5] = new Point3f(1f, 1f, 0f);
        coords[6] = new Point3f(-1f, 0f, 0f);
        coords[7] = new Point3f(0f, 0f, 0f);
        geom.setCoordinates(0, coords);
        int[] indices = { 0, 1, 2, 3, 4, 5, 1, 3, 6, 7};
        geom.setCoordinateIndices(0, indices);
```

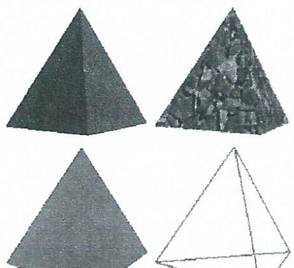


{6, 4}
{0, 1, 2, 3, 4, 5, 6, 7}

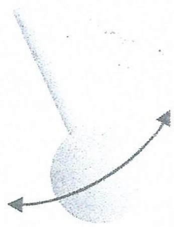
Consider the following appearance configuration.

```
Appearance ap = new Appearance();
ap.setPolygonAttributes(new PolygonAttributes(PolygonAttributes.POLYGON_LINE,
PolygonAttributes.CULL_NONE, 0));
```

2. (3) The figure illustrates 4 possible types of appearance that can be defined in Java 3D. Identify each type and draw the corresponding part of a scene graph representative of the implementation of each type (include all objects.



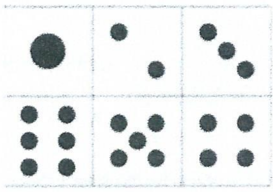
3. (2) Consider that you want to create a 3D scene with a pendulum swinging as shown in the figure. Present the corresponding scene graph with the elements necessary to build the scene. Consider that the pendulum is formed by a cylinder and a sphere. And it has a material-based look. The scene has ambient light and a punctual light as well as a background.



4. (2) Consider that a coordinate point (1, 0, 0) on the surface of an object has a normal in the (0, 1, 0) direction and is illuminated by a light with intensity (1, 1, 0) positioned at (0, 2, 0). The view is positioned at (5, 3, 0). If the specular reflection coefficients of the chosen material for the object's appearance are (0.3, 0.5, 0.2), and the brightness equal to 10, what are the RGB values of specular reflection at that point? It is not necessary to calculate the values, only present the equations.

$$I = I_a k_a + I_p k_d \cos\theta + I_p k_s \cos^n a$$

5. (2) Define a TexCoord2f [] array with the unique texture coordinates, so as to map the 6 pieces of the image on the right, on the faces of a cube.



6. (3) Complete the following code that creates a scene with 2 objects rotating around the Y axis. Consider that there are 2 functions called createGeometry () and createTextureAppearance () that create geometry and appearance for shape2.

```
private BranchGroup createSceneGraph() {

BranchGroup root = new BranchGroup();
TransformGroup spin = new TransformGroup();
spin.setCapability(                                    ); //1°  TransformGroup.
root.addChild(spin);                                              Allow.Transform.write

          shape1 = new Cone(); //2° Cone
Transform3D tr1 = new Transform3D();
tr1.set(new Vector3f(0,0,0.99f), 0.25f);                       → tr1.setRotation
tr1.                                              ); //3°  (new AxisAngle4D(1,1,0,
TransformGroup tg1 = new TransformGroup(tr1);                   Math.PI/2)
spin.addChild(tg1); //4°
tg1.addChild(      ); //5° → Shape1

PolygonAttributes pa = new PolygonAttributes(                              , //6°
  PolygonAttributes.CULL_NONE,0);
pa.setBackFaceNormalFlip(true);
ap.setPolygonAttributes(pa);
Shape3D shape2 = new Shape3D(        ); //7°
Transform3D tr2 = new Transform3D();
tr2.set(new Vector3f(0,0,-0.99f), 0.25f);
tr2.setRotation(new AxisAngle4d(1, 0, 0, Math.PI/2));
TransformGroup tg2 = new TransformGroup(tr2);
spin.addChild(    ); //8° → tg2
tg2.addChild(      ); //9° → Shape 3D

Alpha alpha = new Alpha(-1, 4000);
RotationInterpolator rotator = new RotationInterpolator(      ,      ); //10°
                         //11°

BoundingSphere bounds = new BoundingSphere();
rotator.setSchedulingBounds(bounds);

AmbientLight light = new AmbientLight(true,                           ); //12°
light.setInfluencingBounds(bounds);
root.addChild(light);
PointLight ptlight = new PointLight(new Color3f(Color.green),
                       , new Point3f(1f,0f,0f)); //13°
ptlight.setInfluencingBounds(bounds);
root.addChild(ptlight);

return root;
}
```

7. (3) Complete the following code to configure a **RotPosPathInterpolator** to control the movement of an object along a triangular path formed by vertices A (-2, 0, 2), B (-2, 0, -2), and C (2.0 , 2). The object begins to move at vertex A and should take 2 seconds to move from A to B, 3 seconds from B to C and 2 seconds from C to A. At each vertex of the trajectory, the object must rotate on itself so that it faces the next vertex. All rotations should take 1 second. Assume that the **InterpolatorData** class developed in class is used to make it easier to set up the interpolator.

```
public BranchGroup createSceneGraph() {
  BranchGroup root = new BranchGroup();
  BoundingSphere bounds = new BoundingSphere();
  InterpolatorData iData = new InterpolatorData();

  //Left bottom corner.
  iData.add(new Point3f(-2f, 0f, 2f),        );          //1°, 2°
  //Left upper corner.
  iData.add(new Point3f(-2f, 0f, -2f),  , );       //3°, 4°
  iData.add(new Point3f(-2f, 0f, -2f), ,); //5°, 6°
  //Right bottom corner
  iData.add(new Point3f(2f, 0f, 2f), ,);   //7°, 8°
  iData.add(new Point3f(2f, 0f, 2f), ,);     //9°, 10°
  //Left bottom corner.
  iData.add(new Point3f(-2f, 0f, 2f), -270f,);    //11°
  iData.add(new Point3f(-2f, 0f, 2f), ,);       //12°, 13°

  // Adds the interpolator.
  Alpha alpha = new Alpha(-1,);  //15°
  TransformGroup target = new TransformGroup();
                                                     //16°
  Transform3D axis = new Transform3D();
  RotPosPathInterpolator interpolator = new RotPosPathInterpolator(alpha,
  , axis, iData.getAlphas(), iData.getOrientations(),   //17°
  iData.getPositions());
  interpolator.setSchedulingBounds(bounds);
  root.addChild(    );  //18°
  target.addChild(    );  //19°

  target.addChild(    );  //20°
  return root;
}
```

8. (1) Add the missing code to cause a rotation of 45° along the X-axis to an object that is the child of the **TransformGroup** node obtained through picking.

```
public void mouseClicked(MouseEvent e) {
  pc.setShapeLocation(e);
  PickResult result = pc.pickClosest();

  if(result != null) {
    TransformGroup tg =
    (TransformGroup)(result.getNode(PickResult.TRANSFORM_GROUP))
    if(tg != null) {
       //Código em falta



  }
}
```

9. (1) Explain what the following code is and what it does in concrete.

```
SimpleUniverse su = new SimpleUniverse(cv);
Transform3D tRot = new Transform3D();
tRot.rotX(Math.toRadians(-45));
Transform3D tTra = new Transform3D();
tTra.setTranslation(new Vector3d(0, 0, 11));
tRot.mul(tTra);
su.getViewingPlatform().getViewPlatformTransform().setTransform(tRot);
```

0/

10. (1) Explain why it is necessary to use shadow simulation techniques if you want to produce shadows in Java 3D graphics applications.