



Asignatura: Application Development for Mobile Devices.

Tarea 13: Diseño de figuras geométricas en 3D.

El lenguaje de programación Java contiene las capacidades para realizar muchas tareas. Ya sea que se necesite una imagen desde una biblioteca de Java, que probablemente existe para llevar a cabo su tarea como aplicación de escritorio o, por ejemplo, la creación de imágenes 3D en Java, que también se logra fácilmente a través de la biblioteca de Java 3D. El uso de los objetos contenidos en la biblioteca, puede incluir anuncios de iluminación y efectos de perspectiva a una forma geométrica para crear una imagen 3D.

En este caso, se utiliza código con conceptos básicos de matemáticas, modelando un cubo con una matriz 3D y una matriz de transformación para 2D. El primer objetivo es modelar un cubo y, a partir de éste, modelar un cono, un cilindro, una esfera y otros objetos, como se muestran en la figura 1.

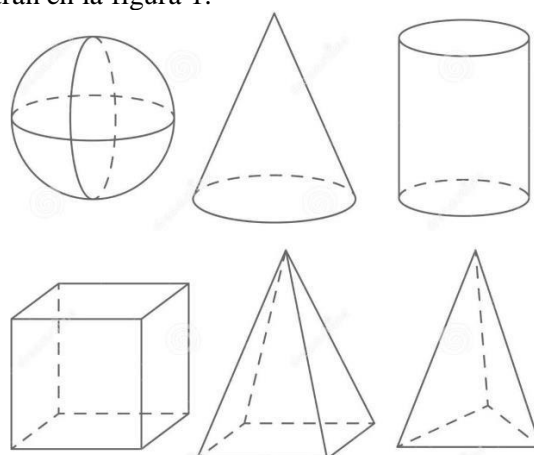


Figura 1. Figuras geométricas obtenidas a partir de un cubo.

EJEMPLO.

Utilizar el siguiente código Java y modificarlo para obtener cada una de las figuras mencionadas.

// La clase GiraCubo que modela un cubo en 3D y se gira con el mouse.

```
import java.awt.*;
import java.awt.event.*;
public class GiraCubo{
    Frame f;
    public GiraCubo(){
        f = new Frame("Cubo en 3D");
        f.setBackground(Color.YELLOW);
        f.addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent e){
                System.exit(0);
            }
        });
        f.setLayout(new BorderLayout());
        f.add("Center", new Perspectiva());
        Dimension dim = f.getToolkit().getScreenSize();
        f.setSize(dim.width/2, dim.height/2);
        f.setLocation(dim.width/4, dim.height/4);
        f.setVisible(true);
    }
    public static void main(String [] args){
        new GiraCubo();
    }
}

class Perspectiva extends Canvas implements MouseMotionListener{ // clase para pintar
    int centerX, centerY, maxX, maxY, minMaxXY;
```



```
Dimension dim;
Obj obj = new Obj();
public Perspectiva(){
    dim = getSize();
    centerX = maxX/2;
    centerY = maxY/2;
    addMouseListener(this);
}
int iX(float x){
    return Math.round(centerX + x);
}
int iY(float y){
    return Math.round(centerY - y);
}
void line(Graphics g, int i, int j){
    Point2D p = obj.vScr[i], q = obj.vScr[j];
    g.drawLine(iX(p.x), iY(p.y), iX(q.x), iY(q.y));
}
public void paint(Graphics g){
    dim = getSize();
    maxX = dim.width-1; maxY = dim.height-1; minMaxXY=Math.min(maxX, maxY);
    centerX = maxX/2;
    centerY = maxY/2;
    obj.d = obj.rho*minMaxXY/obj.objSize;
    obj.eyeAndScreen();
    line(g, 0, 1); line(g, 1, 2); line(g, 2, 3); line(g, 3, 0); // aristas horizontales
inferiores
    line(g, 4, 5); line(g, 5, 6); line(g, 6, 7); line(g, 7, 4); // aristas horizontales
superiores
    line(g, 0, 4); line(g, 1, 5); line(g, 2, 6); line(g, 3, 7); // aristas verticales
}
public void mouseDragged(MouseEvent e){
    obj.theta = (float) getSize().width/e.getX();
    obj.phi = (float) getSize().height/e.getY();
    obj.rho = (obj.phi/obj.theta)*getSize().height;
    centerX = e.getX();
    centerY = e.getY();
    repaint();
}
public void mouseEntered(MouseEvent e){}
public void mouseExited(MouseEvent e){}
public void mouseMoved(MouseEvent e){}
}
class Obj{ // Posee los datos del objeto 3D
    float rho, theta=0.3F, phi=1.3F, d, objSize, v11, v12, v13, v21, v22, v23, v32, v33, v43; //
elementos de la matriz V
    Point3D [] w; // coordenadas universales
    Point2D [] vScr; // coordenadas de la pantalla
    Obj(){ // CAMBIAR LAS COORDENADAS X,Y,Z CON 0,1 PARA CONSTRUIR PRISMA, CILINDRO, PIRAMIDE,
CONO Y ESFERA.
        w = new Point3D[8];
        vScr = new Point2D[8];

        //Cubo
        w[0] = new Point3D(1, -1, -1); // desde la base
        w[1] = new Point3D(1, 1, -1);
        w[2] = new Point3D(-1, 1, -1);
        w[3] = new Point3D(-1, -1, -1);
        w[4] = new Point3D(1, -1, 1);
        w[5] = new Point3D(1, 1, 1);
        w[6] = new Point3D(-1, 1, 1);
        w[7] = new Point3D(-1, -1, 1);
```



```
objSize = (float) Math.sqrt(12F); // = sqrt(2*2 + 2*2 + 2*2) es la distancia entre dos
vertices opuestos
rho = 5*objSize; // para cambiar la perspectiva
}
void initPersp(){
    float costh = (float)Math.cos(theta), sinth=(float)Math.sin(theta),
    cosph=(float)Math.cos(phi), sinph=(float)Math.sin(phi);
    v11 = -sinth; v12 = -cosph*costh; v13 = sinph*costh;
    v21 = costh; v22 = -cosph*sinth; v23 = sinph*sinth;
    v32 = sinph; v33 = cosph; v43 = -rho;
}
void eyeAndScreen(){
    initPersp();
    for(int i=0; i<8; i++){
        Point3D p = w[i];
        float x = v11*p.x + v21*p.y, y = v12*p.x + v22*p.y + v32*p.z, z = v13*p.x + v23*p.y +
v33*p.z + v43;
        vScr[i] = new Point2D(-d*x/z, -d*y/z);
    }
}
}
class Point2D{
    float x, y;
    Point2D(float x, float y){
        this.x = x;
        this.y = y;
    }
}
class Point3D{
    float x, y, z;
    Point3D(double x, double y, double z){
        this.x = (float) x;
        this.y = (float) y;
        this.z = (float) z;
    }
}
}
```

EJERCICIO 1.

Convertir el ejemplo anterior `GiraCubo.java` en una aplicación móvil.

EJERCICIO 2.

Utilizando el ejercicio 1, mostrar las figuras indicadas en la figura 1 en una aplicación móvil.

NOTA. Obtener cada una de las figuras mostradas en la figura 1. Insertar las imágenes de las figuras en un documento y guardarlo con la sintaxis `AlumnoTarea13Grupo.pdf` para enviarlo al sitio indicado por el profesor.