



# **Instituto Politécnico Nacional**

## **Escuela Superior de Computo**



Nombre del Alumno:

- Huerta Cortes Alan Antonio

Asignatura: Application Development For Mobile Devices

Profesor: Cifuentes Álvarez Alejandro Sigfrido

Reporte de figuras geométricas

Fecha de entrega: 25 de febrero del 2019

Grupo: 3CV5

# Índice

<b>1.Objetivo.....</b>	<b>3</b>
<b>2.Conceptos previos.....</b>	<b>3</b>
2.1 Canvas.....	3
<b>3.Desarrollo.....</b>	<b>4</b>
<b>4.Conclusiones.....</b>	<b>6</b>
<b>5.Bibliografia.....</b>	<b>6</b>

## **1.Objetivo:**

Realizar un proyecto que con ayuda de la clase Canvas dibuje figuras geométricas en 3D las cuales poseerán de movimiento de 360 grados todo esto tomando como referencia el cubo para su correcta realización.

## **2.Conceptos previos**

### **2.1Canvas**

Es la clase base para escribir aplicaciones que necesitan manejar eventos a bajo nivel, así como realizar peticiones para dibujar gráficos en la pantalla. Los videojuegos, por ejemplo, hacen un uso intenso y extenso de la clase Canvas. Desde la perspectiva de un desarrollador de aplicaciones, la clase Canvas es totalmente intercambiable con las clases estándar de tipo Screen, por lo que un Midlet puede mezclar y asociar pantallas tipo Canvas con pantallas de alto nivel (Form, List, etc) si así lo precisa. Canvas proporciona una serie de métodos a los desarrolladores para manejar acciones de juego, eventos de teclado y eventos de puntero (si el dispositivo los soporta). También se proporcionan métodos para identificar las capacidades gráficas del dispositivo y para asociar las teclas o botones con acciones de juego. Los eventos de teclado se notifican en relación con unos key codes, que están directamente asociados con teclas concretas del dispositivo, y cuyo uso puede perjudicar la portabilidad de las aplicaciones. Los Midlets portables deben usar acciones de juego en lugar de key codes. Al igual que otras subclases de Displayable, la clase Canvas permite a una aplicación registrar un listener para los comandos. Sin embargo, a diferencia de otros Displayables, la clase Canvas requiere que las aplicaciones sean subclases de ésta, en lugar de simplemente usarla. El método paint() se declara como abstracto, por lo que las aplicaciones deben proporcionar una implementación para el mismo. Otros métodos de notificación de eventos no están declarados como abstractos, pero su implementación por defecto está vacía, de modo que las aplicaciones pueden redefinir sólo aquellos en los que estén interesadas.

### 3. Desarrollo

El desarrollo de este proyecto consiste en crear figuras geométricas con movimiento a partir de la unión de sus vértices.

1. A continuación, se muestra el código para la construcción de las figuras geométricas a partir de sus vértices.

```
3 class Obj{
4     float rho, theta=0.3F, phi=1.3F, d, objSize, v11, v12, v13, v21, v22, v23, v32, v33, v43; // elementos de la matriz V
5     Point3D [] w; // coordenadas universales
6     Point2D [] vScr; // coordenadas de la pantalla
7     Obj(){ // CAMBIAR LAS COORDENADAS X,Y,Z CON 0,1 PARA CONSTRUIR PRISMA, CILINDRO, PIRAMIDE, CONO Y ESFERA.
8
9         w = new Point3D[8];
10        vScr = new Point2D[8];
11
12        //Cubo
13        w[0] = new Point3D(0, 0, -2); // 1, -1, -1 desde la base 0, 0, -2
14        w[1] = new Point3D(0, 0, -2); // 1, 1, -1
15        w[2] = new Point3D(0, 0, -2); // -1, 1, -1
16        w[3] = new Point3D(0, 0, -2); // -1, -1, -1
17        w[4] = new Point3D(1, -1, 1); // 1, -1, 1
18        w[5] = new Point3D(1, 1, 1); // 1, 1, 1
19        w[6] = new Point3D(-1, 1, 1); // -1, 1, 1
20        w[7] = new Point3D(-1, -1, 1); // -1, -1, 1
21        objSize = (float) Math.sqrt(12F); // = sqrt(2*2 + 2*2 + 2*2) es la distancia entre dos vertices opuestos
22        rho = 5 * objSize; // para cambiar la perspectiva
23    }
24    void initPersp(){
25        float costh = (float) Math.cos(theta), sinth=(float) Math.sin(theta), cosph=(float) Math.cos(phi), sinph=(float) Math.sin(phi);
26        v11 = -sinth; v12 = -cosph*costh; v13 = sinph*costh;
27        v21 = costh; v22 = -cosph*sinth; v23 = sinph*sinth;
28        v32 = sinph; v33 = cosph; v43 = -rho;
29    }
30
31    void eyeAndScreen(){
32        initPersp();
33        for(int i=0; i<8; i++){
34            Point3D p = w[i];
35            float x = v11*p.x + v21*p.y + v31*p.z, y = v12*p.x + v22*p.y + v32*p.z, z = v13*p.x + v23*p.y + v33*p.z + v43;
36            vScr[i] = new Point2D(-d*x/z, -d*y/z);
37        }
38    }
39 }
```

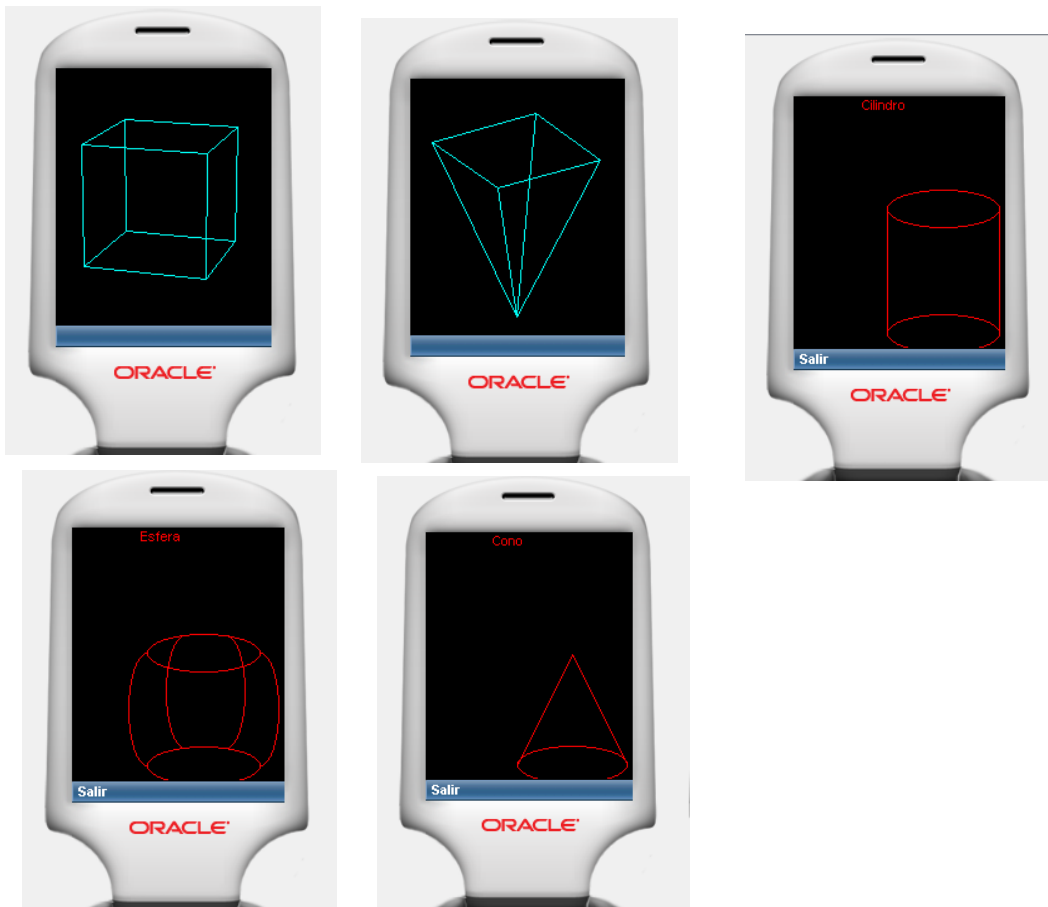
2. Código que dibuja en el midlet

```
47 public void paint(Graphics g) {
48     g.setColor(0,0,0);
49     g.fillRect(0, 0, getWidth(),getHeight());
50     g.setColor(3,252,244);
51     maxX=getWidth()-1; maxY=getHeight()-1; minMaxXY=Math.min(maxX,maxY);
52     centerX = maxX/2;
53     centerY = maxY/2;
54     obj.d = obj.rho*minMaxXY/obj.objSize;
55     obj.eyeAndScreen();
56     line(g, 0, 1); line(g, 1, 2); line(g, 2, 3); line(g, 3, 0); // aristas h
57     line(g, 4, 5); line(g, 5, 6); line(g, 6, 7); line(g, 7, 4); // aristas h
58     line(g, 0, 4); line(g, 1, 5); line(g, 2, 6); line(g, 3, 7); // aristas v
59 }
```

3. Código que les da el movimiento a las figuras dentro del canvas.

```
61 public void keyPressed(int keyCode){ // sustituye la
62     if(keyCode == getKeyCode(DOWN))
63         y-=2;
64     if(keyCode == getKeyCode(RIGHT))
65         x-=2;
66     if(keyCode == getKeyCode(LEFT))
67         x+=2;
68     if(keyCode == getKeyCode(UP))
69         y+=2;
70     obj.theta = (float) getWidth()/x;
71     obj.phi = (float) getHeight()/y;
72     obj.rho = (obj.phi/obj.theta)*getHeight();
73     centerX = x;
74     centerY = y;
75     repaint();
}
```

4. Ejecución del proyecto realizado.



## **4.Conclusiones**

Se realizo la practica propuesta en clase de forma satisfactoria y aprendí un poco mas el uso de canvas y como realizar trazos sobrescribiendo el método Paint que pertenece a la clase de canvas así como también dibujar figuras 3D partiendo de sus vértices y pudiendo manipular el movimiento de ellas con solo aumentar o disminuir las coordenadas.

## **5.Bibliografia**

<http://leo.ugr.es/J2ME/MIDP/graficosbajo.htm>