

# JAVA 2D

Luis Gajardo Díaz  
*lgajardo@ubiobio.cl*

## ¿Qué es Java2D?

- Es una librería java para gráficos en 2 dimensiones (2D).
- Fue desarrollada por Sun Microsystem (ahora Oracle).
  - <http://java.sun.com/products/java-media/2D/index.jsp>
- Amplía muchas de las capacidades gráficas de la biblioteca AWT (*Abstract Window Toolkit - Herramientas Abstractas de Ventanas*)
- El rango que abarcan todas estas mejoras es muy amplio:
  - Renderizado
  - Definición de figuras geométricas
  - Uso de fuentes de letras
  - Enriquecimiento en la definición del color
  - Manipulación de imágenes

## ¿Qué es Java2D?

- Permite la creación de bibliotecas de gráficos avanzados, por ejemplo:
  - **JMF** (*Java Media Framework - Entorno de Trabajo* de Java para Medios Audiovisuales)
  - **Java 3D** (java para el entorno 3D).
- La clase `java.awt.Graphics2D` extiende a `java.awt.Graphics`.
- **Graphics2D** proporciona un control más potente sobre la presentación de texto, imágenes o figuras geométricas.
- Un objeto **Graphics** representa el lienzo abstracto donde dibujar cualquier cosa.
- Este lienzo puede estar enlazado con un área física de un monitor, o representar una imagen en memoria que sólo se desea manipular y no tiene representación directa durante este proceso.

3

## Renderizado de imágenes con Graphics2D

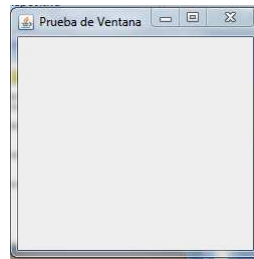
- Dibujar gráficos dentro de ventanas

```
import javax.swing.*;

public class Ventana extends JFrame {

    public Ventana() {
        super ("Prueba de Ventana");
        this.setSize(250,250);
        this.setDefaultCloseOperation(EXIT_ON_CLOSE);
        this.setVisible(true);
    }

    public static void main (String [] args) {
        Ventana v = new Ventana();
    }
}
```



4

## Renderizado de imágenes con Graphics2D

- Dibujar gráficos a pantalla completa

```
import javax.swing.JFrame;
import java.awt.*;

public class Ventana2 extends JFrame {

    public Ventana2() {
        // resolución de 800x600 a 16 bits, 60 hertz
        DisplayMode displayMode = new DisplayMode(800, 600, 16, 60);

        // obtener el tipo de dispositivo grafico (GraphicsDevice)
        GraphicsEnvironment environment = GraphicsEnvironment.getLocalGraphicsEnvironment();
        GraphicsDevice device = environment.getDefaultScreenDevice();

        // usar el JFrame como una ventana a pantalla completa
        device.setFullScreenWindow(this);

        // cambiar el modo de despliegue
        device.setDisplayMode(displayMode);
    }

    public static void main(String[] arg) {
        new Ventana2();
        System.out.println("He creado la ventana");
    }
}
```

5

## El color en Java2D

- Crear un nuevo color (RGB)

```
Graphics2D g2 = (Graphics2D)g;
Color color = new Color(200, 100, 50);
g2.setColor(color);
...
```

**Color(int r, int g, int b)**

- r: componente rojo [0..255]
- g: componente verde [0..255]
- b: componente azul [0..255]

- Colores predefinidos

Color predefinido	Valor del color
Color.blue	new Color(0,0,255)
Color.green	new Color(0,255,0)
Color.red	new Color(255,0,0)
Color.orange	new Color(255,200,0)
Color.yellow	new Color(255,255,0)
Color.black	new Color(0,0,0)
Color.white	new Color(255,255,255)
Color.grey	new Color(128,128,128)
Color.cyan	new Color(0,255,255)
Color.magenta	new Color(255,0,255)
Color.pink	new Color(255,175,175)
Color.darkGray	new Color(64,64,64)
Color.lightGray	new Color(192,192,192)

```
Graphics2D g2 = (Graphics2D)g;
g2.setColor(Color.red);
...
//las figuras saldrán en color azul
//de aquí en adelante
```

6

## Figuras geométricas en Java2D

- Dibujar un rectángulo

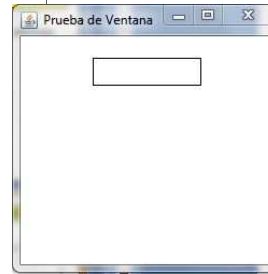
```
import javax.swing.*;
import java.awt.*;
import java.awt.geom.*;

public class Grafical extends JFrame {

    public Grafical() {
        super ("Prueba de Ventana");
        this.setSize(250,250);
        this.setDefaultCloseOperation(EXIT_ON_CLOSE);
        this.setVisible(true);
    }

    public static void main (String [] args) {
        Grafical v = new Grafical();
    }

    public void paint (Graphics g) {
        //parametros Rectangle2D: x, y, ancho, alto
        Rectangle2D r2 = new Rectangle2D.Float(75, 50, 100, 25);
        Graphics2D g2 = (Graphics2D)g;
        g2.draw(r2);
    }
}
```

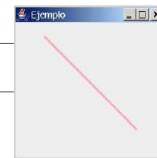


7

## Figuras geométricas en Java2D

- Dibujar una línea

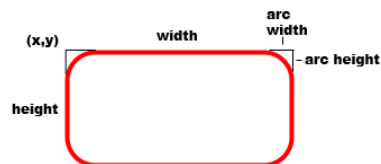
```
//parametros Line2D: x1, y1, x2, y2
Line2D l = new Line2D.Float(50, 50, 200, 200);
```



- Dibujar un rectángulo con borde redondeados

- Los parámetros pueden ser tipo float o double

```
//parametros RoundRectangle2D: x, y, ancho, alto, anchura del arco, altura del arco
RoundRectangle2D q = new RoundRectangle2D.Float(50.0f, 50.0f, 100.0f, 150.0f,
25.0f, 25.0f);
```

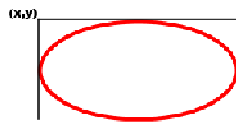


8

## Figuras geométricas en Java2D

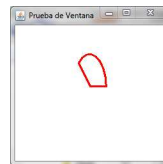
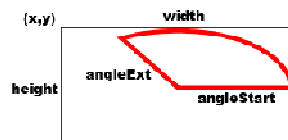
- Dibujar una elipse

```
//parametros Ellipse2D: x, y, ancho, alto
Ellipse2D e = new Ellipse2D.Float(100.0f,75.0f,50.0f,100.0f);
```



- Dibujar un arco

```
//parametros Arc2D: x, y, ancho, alto, inicio angulo, extension angulo, tipo arco
Arc2D a = new Arc2D.Float(100.0f, 75.0f, 50.0f, 100.0f, 0.0f, 135.0f, Arc2D.PIE);
```

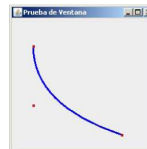
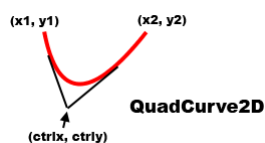


9

## Figuras geométricas en Java2D

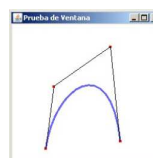
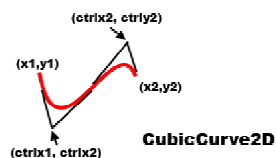
- Dibujar una curva cuadrática de Bézier (de grado 2)

```
//parametros QuadCurve2D: x1, y1, control x, control y, x2, y2
QuadCurve2D q = new QuadCurve2D.Float(40.0f, 70.0f, 40.0f, 170.0f, 190.0f, 220.0f);
```



- Dibujar una curva cuadrática de Bézier (de grado 3)

```
//parametros CubicCurve2D: x1, y1, control x1, control y1, control x2, control y2, x2, y2
CubicCurve2D c=new CubicCurve2D.Float(40.0f, 60.0f, 60.0f, 120.0f, 140.0f, 130.0f, 150.0f, 210.0f);
```



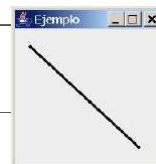
10

## Figuras geométricas en Java2D

### • Usando Puntos

- La clase Point2D no dibuja nada, sino que es la representación de los puntos en Java2D.
- Sin embargo, sí se pueden dibujar figuras a partir de puntos.
- Todos los constructores de figuras que hemos visto anteriormente están sobrecargados para poder construir figuras tanto a partir de coordenadas sueltas como a partir de objetos Point2D.

```
//crear 2 puntos y luego en base a ellos una linea
Point2D p1 = new Point2D.Float(23.5f, 48.9f);
Point2D p2 = new Point2D.Float(158.0f, 173.0f);
Line2D l = new Line2D.Float(p1, p2);
```



11

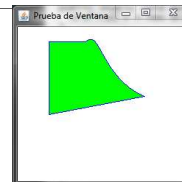
## Figuras geométricas en Java2D

### GeneralPath

- La clase GeneralPath permite definir una figura mediante una secuencia de trazos que pueden establecerse a través de las siguientes funciones:

Método	Descripción
GeneralPath (constructor)	Representa una figura geométrica construida a partir de líneas rectas y de curvas cuadráticas y cúbicas
moveTo	Desplaza la punta del lápiz hacia un punto concreto
curveTo	Describe una curva cúbica desde un punto inicial a uno final
lineTo	Define una línea recta desde un punto inicial a uno final
quadTo	Representa una curva cuadrática desde un punto inicial a uno final
closePath	Cierra el trazado de la figura uniéndolo mediante una línea recta la posición actual del lápiz con la del inicio del dibujo

```
//Creacion del GeneralPath
g2.setColor(Color.green);
GeneralPath gp = new GeneralPath();
gp.moveTo(50.0f,50.0f);
gp.lineTo(100.0f,50.0f);
gp.curveTo(120.0f,30.0f,120.0f,100.0f,180.0f,125.0f);
gp.lineTo(50.0f,150.0f);
gp.closePath();
//Relleno y contorno
g2.fill(gp);
g2.setColor(Color.blue);
```



12

## Modificación del contexto de Graphics2D

- Especificar el estilo de línea

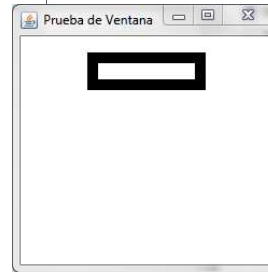
```
import javax.swing.*;
import java.awt.*;
import java.awt.geom.*;

public class Contexto1 extends JFrame {

    public Contexto1() {
        super ("Prueba de Ventana");
        this.setSize(250,250);
        this.setDefaultCloseOperation(EXIT_ON_CLOSE);
        this.setVisible(true);
    }

    public static void main (String [] args) {
        Contexto1 v = new Contexto1();
    }

    public void paint (Graphics g) {
        Graphics2D g2 = (Graphics2D)g;
        Rectangle2D r2 = new Rectangle2D.Float(75,50,100,25);
        //indicar el ancho de la linea
        Stroke pincel = new BasicStroke(10.0f);
        g2.setStroke(pincel);
        g2.draw(r2);
    }
}
```



13

## Especificación de los atributos de relleno

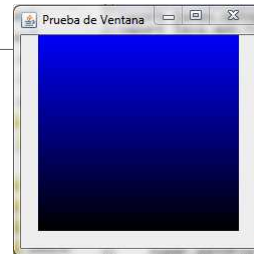
- Gradiente de color

```
import javax.swing.*;
import java.awt.*;
import java.awt.geom.*;

public class Relleno1 extends JFrame {

    ...

    public void paint (Graphics g) {
        super.paint(g);
        Graphics2D g2 = (Graphics2D)g;
        GradientPaint gp1 =
            new GradientPaint(50.0f,25.0f,Color.blue, 50.0f, 225.0f, Color.black);
        g2.setPaint(gp1);
        Rectangle2D r1 = new Rectangle2D.Float(25, 25, 200, 200);
        g2.fill(r1);
        GradientPaint gp2 =
            new GradientPaint(250.0f,25.0f,Color.blue, 450.0f, 225.0f, Color.black);
        g2.setPaint(gp2);
        Rectangle2D r2 = new Rectangle2D.Float(250, 25, 200, 200);
        g2.fill(r2);
    }
}
```



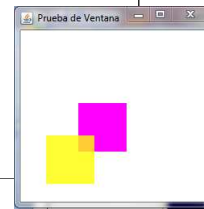
14

## Especificación de los atributos de relleno

- **El canal alfa**

- El canal alfa hace referencia a un componente que poseen todos los colores y que corresponde a la transparencia. De esta forma es posible definir figuras con distintos grados de transparencia.

```
public void paint (Graphics g) {
    Graphics2D g2 = (Graphics2D)g;
    g2.setColor(Color.white);
    Rectangle2D r0 = new Rectangle2D.Float(0,0,250,250);
    g2.fill(r0);
    //parametros: tipo: SRC_OVER = predomina color figura de arriba, 0.5 = 50%
    AlphaComposite ac = AlphaComposite.getInstance(AlphaComposite.SRC_OVER, 0.5f);
    g2.setColor(Color.magenta);
    //realiza la composicion de colores
    g2.setComposite(ac);
    Rectangle2D r3 = new Rectangle2D.Float(80,120,60,60);
    g2.fill(r3);
    ac = AlphaComposite.getInstance(AlphaComposite.SRC_OVER, 0.8f);
    g2.setColor(Color.yellow);
    g2.setComposite(ac);
    Rectangle2D r4 = new Rectangle2D.Float(40,160,60,60);
    g2.fill(r4);
}
```

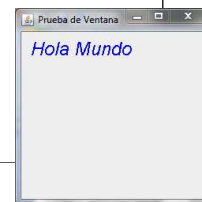


15

## Tratamiento de texto con Java2D

- **Escribir una línea de texto**

```
public void paint (Graphics g) {
    super.paint(g);
    Graphics2D g2 = (Graphics2D)g;
    g2.setColor(Color.blue);
    //selecciona la fuente activa
    g2.setFont(new Font("Arial", Font.ITALIC, 24));
    //parametros: texto, x, y
    g2.drawString("Hola Mundo", 20, 60);
}
```



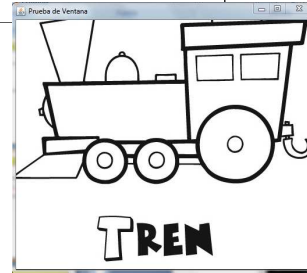
16



## Imágenes con Java2D

- Cargar una imagen desde disco

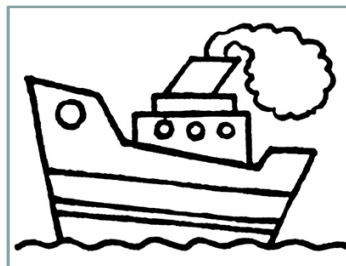
```
public void paint(Graphics g) {  
    Graphics2D g2 = (Graphics2D)g;  
    //parametro: ruta y nombre del archivo a cargar  
    Image im = Toolkit.getDefaultToolkit().getImage("tren.jpg");  
    g2.drawImage(im, 0, 20, this);  
}
```



17

## EJERCICIO - LABORATORIO

- Implementar una aplicación en Java que permita dibujar en una ventana de 800x600 pixeles la siguiente figura:



- Debe utilizar únicamente las primitivas geométricas de Java 2D
- Escoja los colores que desee
- Entrega del código fuente por PVA.
- Trabajo individual

18