¿Para que sirve un JPanel?

- Es un contenedor genérico
  - Puede contener objetos SWING o AWT
  - Orden según “capas”.
- Por ejemplo, podemos:
  - Dibujar (Utilizando Java 2D API)
  - Mostrar imágenes
- Metodos importantes
  - paint(Graphics g)
    - Podemos sobreescribir este método
  - repaint()
  - setLayout(LayoutManager m)
  - add(Component item, Object Constraints)

Dibujando en un JPanel

- Sobreescribimos el metodo paint()
- Utilizamos la clase Graphics2D
- Clase RenderingHints
  - Pares clave-valor
- Posibilidad de acceder a las dimensiones del panel "desde dentro"
  - Clase Dimension
- Multitud de objetos para crear formas
  - Ejemplo Ellipse2D
- Necesario definir parametros de trazo, color y relleno
- Uso del método draw() para plasmar el gráfico


TestGeneral.jav
Board.java
BoardImage.java
Anir
package es.uned.inicio;
import javax.swing.JFrame;
public class TestGeneral extends JFrame {
public TestGeneral() {
//add(new Board());
//this.add(new BoardImage());
//this.add(new Animacion());
//this.add(new Animacion2());
//this.add(new Animacion3());
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setSize(360, 310);
setLocationRelativeTo(null);
setTitle("Test");
setVisible(true);
}
public static void main(String[] args) {
new TestGeneral();
}
}

```java
package es.uned.inicio.paneles;

import java.awt.BasicStroke;

public class Board extends JPanel{
```

```java
public void paint(Graphics g)
{
  super.paint(g);

  Graphics2D g2 = (Graphics2D) g;

  RenderingHints rh =
        new RenderingHints(RenderingHints.KEY_ANTIALIASING,
                           RenderingHints.VALUE_ANTIALIAS_ON);

  rh.put(RenderingHints.KEY_RENDERING,
         RenderingHints.VALUE_RENDER_QUALITY);

  g2.setRenderingHints(rh);

  Dimension size = getSize();
  double w = size.getWidth();
  double h = size.getHeight();

  System.out.println("Ancho: "+w+" -- Alto: "+h);

  Ellipse2D e = new Ellipse2D.Double(0, 0, 80, 130);
  g2.setStroke(new BasicStroke(1));
  g2.setColor(Color.red);
  g2.setBackground(Color.red);

  g2.draw(e);
  g2.fill(e);

  g2.dispose();
}
```

Imágenes en un JPanel

- La metodologia es la misma.
  - Sobreescribir el método paint() de JPanel.
- Podemos utilizar la clase ImageIcon
  - Recibe un objeto URL que apunta a una imagen
- Variables de instancia de ImageIcon:
  - Image. Contiene la imagen que hemos importado
- Podemos utilizar el método drawImage mostrar la imagen

```java
package es.uned.inicio;

import javax.swing.JFrame;

public class TestGeneral extends JFrame {


    public TestGeneral() {

        add(new Board());

        //this.add(new BoardImage());

        //this.add(new Animacion());

        //this.add(new Animacion2());

        //this.add(new Animacion3());

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(360, 310);
        setLocationRelativeTo(null);
        setTitle("Test");
        setVisible(true);
    }

    public static void main(String[] args) {
        new TestGeneral();
    }
}
```

```java
package es.uned.inicio.paneles;

import java.awt.Graphics;

public class BoardImage extends JPanel {

    Image imagen;

    public BoardImage() {
        ImageIcon ii = new ImageIcon(this.getClass().getResource("../uned.jpg"));
        imagen = ii.getImage();
    }

    public void paint(Graphics g) {

        super.paint(g);
        Graphics2D g2d = (Graphics2D) g;
        g2d.drawImage(imagen, 10, 10, null);


    }
}
```

Animando Cosas

- Control de tiempo
  - Uso de la clase Timer de Swing
    - Lanza un evento ActionPerformed
  - Uso de la clase Timer de java.util
    - Llama a run() de la interfaz TimerTask
  - Otra aproximacion es utilizar Threads
- Clase Toolkit.
  - Implementaciones nativas graficas
  - Es el pegamento que une las clases Java independientes de la plataforma con sus correspondientes en e
- Recomendable el uso de dispose()
  - Borra el contexto gráfico.

```java
package es.uned.inicio;

import javax.swing.JFrame;

public class TestGeneral extends JFrame {


    public TestGeneral() {

        //add(new Board());

        this.add(new BoardImage());

        //this.add(new Animacion());

        //this.add(new Animacion2());

        //this.add(new Animacion3());

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(360, 310);
        setLocationRelativeTo(null);
        setTitle("Test");
        setVisible(true);
    }

    public static void main(String[] args) {
        new TestGeneral();
    }
}
```

```
package es.uned.inicio.animaciones;

import java.awt.Color;

public class Animacion extends JPanel implements ActionListener {

    Image imagen;
    Timer timer;
    int x, y;

    public Animacion() {
        setBackground(Color.BLACK);

        ImageIcon ii =
            new ImageIcon(this.getClass().getResource("../uned2.jpg"));
        imagen = ii.getImage();

        setDoubleBuffered(true);

        x = y = 10;
        timer = new Timer(30, this);
        timer.start();
    }
    public void paint(Graphics g) {
        super.paint(g);

        Graphics2D g2d = (Graphics2D)g;
        g2d.drawImage(imagen, x, y, this);
        Toolkit.getDefaultToolkit().sync();
        g.dispose();
    }
```

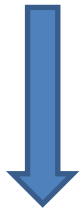
```
    public void actionPerformed(ActionEvent e) {

        x += 1;
        y += 1;

        if (y > 240) {
            y = -45;
            x = -45;
        }
        repaint();
    }
```

Moviendo la nave

por teclado

Modelo (La nave)
Movimiento controlado por teclado

```java
package es.uned.inicio.tecladoMovimiento;

import java.awt.Image;
import java.awt.event.KeyEvent;

import javax.swing.ImageIcon;

public class Craft {

    private String craft = "../uned2.jpg";

    private int dx;
    private int dy;
    private int x;
    private int y;
    private Image image;

    public Craft() {
        ImageIcon ii = new ImageIcon(this.getClass().getResource(craft));
        image = ii.getImage();
        x = 40;
        y = 60;
    }
```



```java
public void move() {
    x += dx;
    y += dy;
}

public int getX() {
    return x;
}

public int getY() {
    return y;
}

public Image getImage() {
    return image;
}
```

Pulsar tecla, soltar tecla

```
public void keyPressed(KeyEvent e) {

    int key = e.getKeyCode();

    if (key == KeyEvent.VK_LEFT) {
        dx = -1;
    }

    if (key == KeyEvent.VK_RIGHT) {
        dx = 1;
    }

    if (key == KeyEvent.VK_UP) {
        dy = -1;
    }

    if (key == KeyEvent.VK_DOWN) {
        dy = 1;
    }
}
```

```
public void keyReleased(KeyEvent e) {
    int key = e.getKeyCode();

    if (key == KeyEvent.VK_LEFT) {
        dx = 0;
    }

    if (key == KeyEvent.VK_RIGHT) {
        dx = 0;
    }

    if (key == KeyEvent.VK_UP) {
        dy = 0;
    }

    if (key == KeyEvent.VK_DOWN) {
        dy = 0;
    }
}
```

Vista (tablero)

```
package es.uned.inicio.tecladoMovimiento;

import java.awt.Color;

public class Tablero extends JPanel implements ActionListener {

    private Timer timer;
    private Craft craft;

    public Tablero() {

        addKeyListener(new TAdapter());
        setFocusable(true);
        setBackground(Color.BLACK);
        setDoubleBuffered(true);

        craft = new Craft();

        timer = new Timer(5, this);
        timer.start();
    }
```

```
public void paint(Graphics g) {
    super.paint(g);

    Graphics2D g2d = (Graphics2D)g;
    g2d.drawImage(craft.getImage(), craft.getX(), craft.getY(), this);

    Toolkit.getDefaultToolkit().sync();
    g.dispose();
}


public void actionPerformed(ActionEvent e) {
    craft.move();
    repaint();
}
```

```
private class TAdapter extends KeyAdapter {

    public void keyReleased(KeyEvent e) {
        craft.keyReleased(e);
    }

    public void keyPressed(KeyEvent e) {
        craft.keyPressed(e);
    }
}
```

Inicio

```
package es.uned.inicio.tecladoMovimiento;

import javax.swing.JFrame;


public class JuegoInicio extends JFrame {

    public JuegoInicio() {

        add(new Tablero());

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(400, 300);
        setLocationRelativeTo(null);
        setTitle("R - Type");
        setResizable(false);
        setVisible(true);
    }

    public static void main(String[] args) {
        new JuegoInicio();
    }
}
```

R-type

Añadimos misil

Misil

video V

```
package es.uned.inicio.disparando;

import java.awt.Image;

public class Misil {

    private int x, y;
    private Image imagen;
    boolean visible;

    private static final int ANCHO_TAB = 390;
    private static final int VEL_MISIL = 2;

    public Misil(int x, int y) {

        ImageIcon ii =
            new ImageIcon(this.getClass().getResource("../missile.jpg"));
        imagen = ii.getImage();
        visible = true;
        this.x = x;
        this.y = y;
    }
```

Misil: métodos

```
public Image getImage() {
    return imagen;
}

public int getX() {
    return x;
}

public int getY() {
    return y;
}

public boolean esVisible() {
    return visible;
}
```

```
public void move() {
    x += VEL_MISIL;
    if (x > ANCHO_TAB)
        visible = false;
}
```

Nave modificada

```
import java.awt.Image;

public class Nave {

    private String craft = "../uned2.jpg";

    private int dx;
    private int dy;
    private int x;
    private int y;
    private Image image;

    private ArrayList<Misil> misilesActivos;

    private static final int TAMANO = 50;

    public Nave() {

        ImageIcon ii = new ImageIcon(this.getClass().getResource(craft));
        image = ii.getImage();
        misilesActivos = new ArrayList<Misil>();
        x = 40;
        y = 60;
    }


    public void move() {
        x += dx;
        y += dy;
```

```java
public Nave() {

    ImageIcon ii = new ImageIcon(this.getClass().getResource(craft));
    image = ii.getImage();
    misilesActivos = new ArrayList<Misil>();
    x = 40;
    y = 60;
}


public void move() {
    x += dx;
    y += dy;
}

public int getX() {
    return x;
}

public int getY() {
    return y;
}
```

```
public int getX() {
    return x;
}

public int getY() {
    return y;
}

public Image getImage() {
    return image;
}

public ArrayList<Misil> getMissiles() {
    return misilesActivos;
}
```

```
public void keyReleased(KeyEvent e) {
    int key = e.getKeyCode();

    if (key == KeyEvent.VK_LEFT) {
        dx = 0;
    }

    if (key == KeyEvent.VK_RIGHT) {
        dx = 0;
    }

    if (key == KeyEvent.VK_UP) {
        dy = 0;
    }

    if (key == KeyEvent.VK_DOWN) {
        dy = 0;
    }
```

```
public void keyPressed(KeyEvent e) {

    int key = e.getKeyCode();

    if (key == KeyEvent.VK_SPACE) {
        disparar();
    }

    if (key == KeyEvent.VK_LEFT) {
        dx = -1;
    }

    if (key == KeyEvent.VK_RIGHT) {
        dx = 1;
    }

    if (key == KeyEvent.VK_UP) {
        dy = -1;
    }

    if (key == KeyEvent.VK_DOWN) {
        dy = 1;
    }
}
```

```
ublic void disparar() {
    misilesActivos.add(new Misil(x + TAMANO, y + TAMANO/2));
```

Tablero2

```
import java.awt.Color;

public class Tablero2 extends JPanel implements ActionListener {

    private Timer timer;
    private Nave craft;

    public Tablero2() {

        addKeyListener(new TAdapter());
        setFocusable(true);
        setBackground(Color.BLACK);
        setDoubleBuffered(true);

        craft = new Nave();

        timer = new Timer(5, this);
        timer.start();
    }
```

Método paint

```
public void paint(Graphics g) {
    super.paint(g);

    Graphics2D g2d = (Graphics2D)g;
    g2d.drawImage(craft.getImage(), craft.getX(), craft.getY(), this);

    ArrayList<Misil> ms = craft.getMissiles();

    for (int i = 0; i < ms.size(); i++ ) {
        Misil m = (Misil) ms.get(i);
        g2d.drawImage(m.getImage(), m.getX(), m.getY(), this);
    }

    Toolkit.getDefaultToolkit().sync();
    g.dispose();
}
```

Clase actionPerformed

```
public void actionPerformed(ActionEvent e) {
    ArrayList<Misil> ms = craft.getMissiles();

    for (int i = 0; i < ms.size(); i++) {
        Misil m = (Misil) ms.get(i);
        if (m.esVisible())
            m.move();
        else ms.remove(i);
    }

    craft.move();
    repaint();
}
private class TAdapter extends KeyAdapter {

    public void keyReleased(KeyEvent e) {
        craft.keyReleased(e);
    }

    public void keyPressed(KeyEvent e) {
        craft.keyPressed(e);
    }
}
```

Clase JuegoInicio

```
package es.uned.inicio.disparando;

import javax.swing.JFrame;

public class JuegoInicio2 extends JFrame {

    public JuegoInicio2() {

        add(new Tablero2());

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(400, 300);
        setLocationRelativeTo(null);
        setTitle("R - Type");
        setResizable(false);
        setVisible(true);
    }

    public static void main(String[] args) {
        new JuegoInicio2();
    }
}
```

Colisiones

Video VI

```
public void checkCollisions() {

    Rectangle r3 = craft.getBounds();

    for (int j = 0; j<aliens.size(); j++) {
        Alien a = (Alien) aliens.get(j);
        Rectangle r2 = a.getBounds();

        if (r3.intersects(r2)) {
            craft.setVisible(false);
            a.setVisible(false);
            ingame = false;
        }
    }

    ArrayList ms = craft.getMissiles();

    for (int i = 0; i < ms.size(); i++) {
        Missile3 m = (Missile3) ms.get(i);

        Rectangle r1 = m.getBounds();

        for (int j = 0; j<aliens.size(); j++) {
            Alien a = (Alien) aliens.get(j);
            Rectangle r2 = a.getBounds();

            if (r1.intersects(r2)) {
                m.setVisible(false);
                a.setVisible(false);
            }
        }
```

```
public Rectangle getBounds() {
    return new Rectangle(x, y, width, height);
}
```

```
public Craft3() {
    ImageIcon ii = new ImageIcon(this.getClass().getResource(craft));
    image = ii.getImage();
    width = image.getWidth(null);
    height = image.getHeight(null);
    missiles = new ArrayList();
    visible = true;
    x = 40;
    y = 60;
}
```