GRAU D'ENGINYERIA INFORMÀTICA

PROGRAMACIÓ II

Bloc 3:

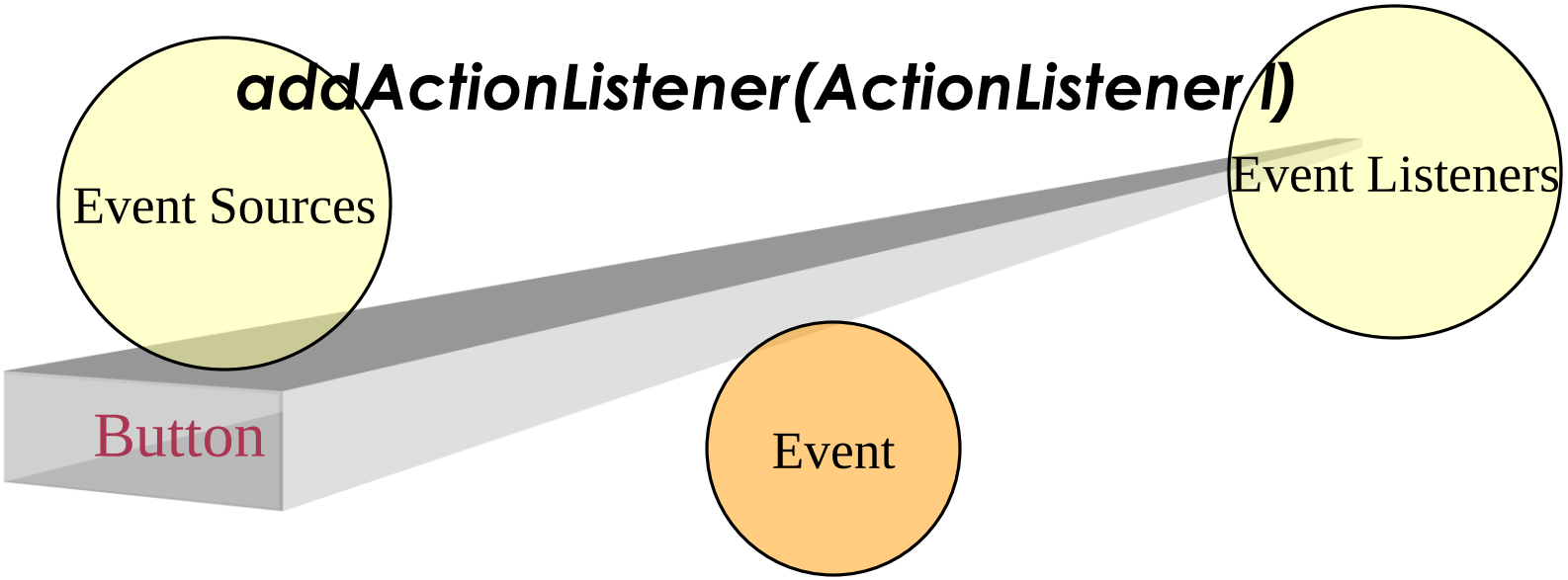
Programació Orientada a Events (4)

Laura Igual
Departament de Matemàtica Aplicada i Anàlisi
Facultat de Matemàtiques
Universitat de Barcelona

REPÀS

Repàs

- Model d'events:



Repàs

- Resum dels passos que es poden seguir per a crear una aplicació interactiva senzilla orientada a events, amb interfície gràfica d'usuari:

1. Determinar els **components** que van a constituir la interfície d'usuari (botons, caixes de text, menús, etc.).
2. Crear una classe per a l'aplicació que contingui la funció **main**().
3. Crear una classe **Finestra**, que hereti de **JFrame**, i que respongui als events
4. La funció **main**() haurà de crear un objecte de la classe **Finestra** i mostrar-la per pantalla amb el tamany i posició adequades.
5. Afegir a l'objecte **Finestra** tots els components que ha de contenir.
6. Definir els objectes **Listener** (objectes que s'ocuparan de respondre als events) per a cada un dels events que han d'estar suportats. Les classes d'aquests objectes implementen les diferents interfícies Listener.
8. Finalment, s'han d'implementar els mètodes de les interfícies **Listener** que es faran càrrec de la gestió dels events.

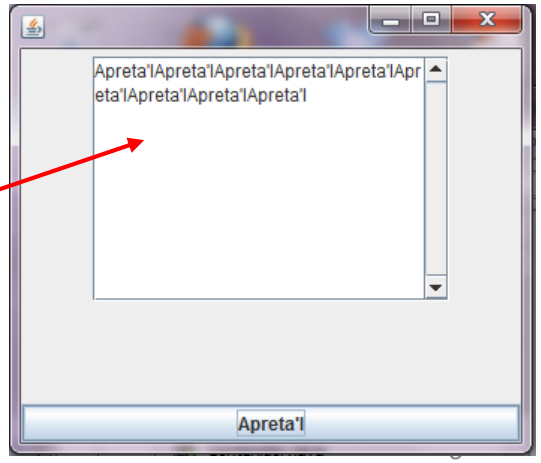
Exemple 2

- Implementeu una interfície gràfica d'usuari que contingui un botó i una àrea de text. Quan es prem el botó s'ha d'escriure el text contingut en ell a l'àrea de text.

```
public class Exemple2 implements ActionListener{
    JTextArea text;
    JButton boto;
    public static void main (String [] args){
       ExempleJTextArea gui  = new ExempleJTextArea();
       gui.go();
    }
    public void go(){
       JFrame frame = new JFrame();
       JPanel panel = new JPanel();
       boto = new JButton("Apreta'l");
       boto.addActionListener(this);
       text = new JTextArea(10,20);
       text.setLineWrap(true);
       JScrollPane scroller = new JScrollPane(text);
       scroller.setVerticalScrollBarPolicy(ScrollPaneConstants.VERTICAL_SCROLLBAR_ALWAYS);
       scroller.setHorizontalScrollBarPolicy(ScrollPaneConstants.HORIZONTAL_SCROLLBAR_NEVER);
       panel.add(scroller);
       frame.getContentPane().add(BorderLayout.CENTER, panel);
       frame.getContentPane().add(BorderLayout.SOUTH, boto);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
       frame.setSize(350,300);
       frame.setVisible(true);       }
    public void actionPerformed(ActionEvent ev){
            text.append(boto.getText());
    } } // Fi clase ExempleJTextArea
```

La declaració del botó s'ha de fer aquí

```
import java.awt.BorderLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;
import javax.swing.ScrollPaneConstants;
```



JList

- Constructor:

  String [] entradesLlista = {“alpha”, “beta”, “gamma”};

  JList list = new JList(entradesLlista);

- Afegir-li una scrollbar vertical:

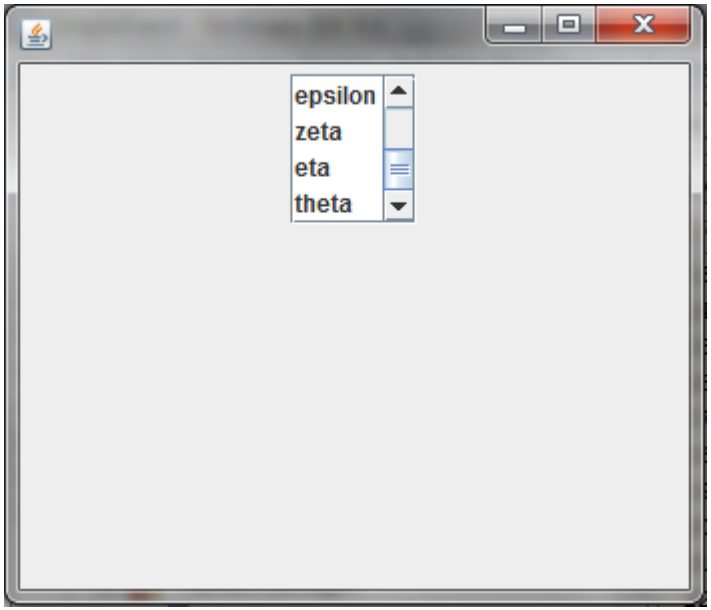
  JScrollPane scroller = new JScrollPane (list);

  scroller.setVerticalScrollBarPolicy(ScrollPaneConstants.VERTICAL_SCROLLBAR_ALWAYS);

  scroller.setHorizontalScrollBarPolicy(ScrollPaneConstants.HORIZONTAL_SCROLLBAR_NEVER);

Exemple 3

- Implementeu una interfície gràfica d'usuari que conté una llista amb els Strings: "alpha", "beta", "gamma", "delta", "epsilon", "zeta","eta", "theta" i que quan seleccionem algun dels elements d'aquesta llista s'imprimeix per pantalla.



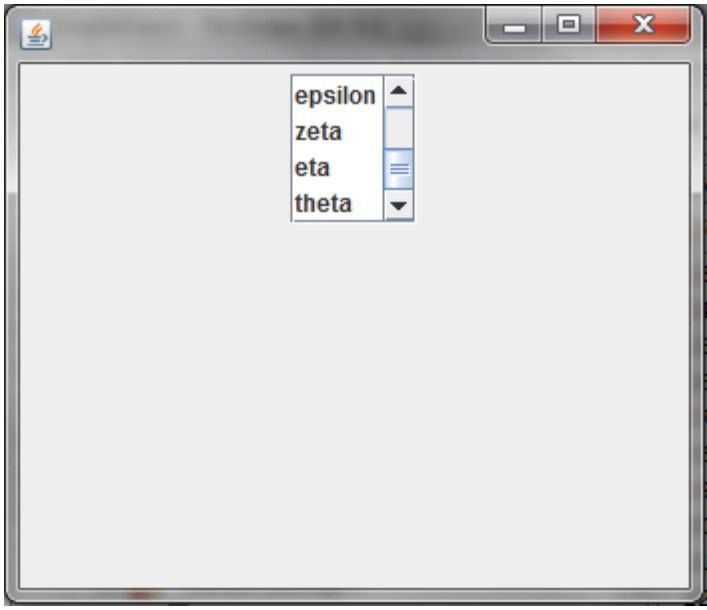
alpha
beta
alpha
gamma
delta
alpha

Exemple 3

```
import javax.swing.JFrame;
import javax.swing.JList;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.ListSelectionModel;
import javax.swing.ScrollPaneConstants;
import javax.swing.event.ListSelectionEvent;
import javax.swing.event.ListSelectionListener;
```

Exemple 3

```
public class ExempleJList implements ListSelectionListener{
   String [] entradesLlista = {"alpha", "beta", "gamma", "delta", "epsilon", "zeta","eta", "theta"};
   JList list  = new JList(entradesLlista );
   public static void main (String [] args){
      ExempleJList gui  = new ExempleJList ();
      gui.go(); }
   public void go(){
      JScrollPane scr = new JScrollPane(list);
      JPanel panel = new JPanel();
      JFrame frame = new JFrame();
      scr.setVerticalScrollBarPolicy(ScrollPaneConstants.VERTICAL_SCROLLBAR_ALWAYS);
      scr.setHorizontalScrollBarPolicy(ScrollPaneConstants.HORIZONTAL_SCROLLBAR_NEVER);
      panel.add(scr);
      list.setVisibleRowCount(4);
      list.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);
      list.addListSelectionListener(this);
      frame.getContentPane().add(BorderLayout.CENTER, panel);
      frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      frame.setSize(350,300);
      frame.setVisible(true);
   }
   public void valueChanged(ListSelectionEvent lse){
      if (lse.getValueIsAdjusting()){
         String selection = (String) list.getSelectedValue();
         System.out.println(selection);    }  }
} // Final classe Exemple
```



Exemple 4: Exercici JOptionPane

- Ompli els forats (indicats amb punts suspensius) del següent codi utilitzant els coneixements previs sobre registre d'events i interfícies Listener, i la següent informació sobre la classes JOptionPane.

JOptionPane

- Els mètodes més importants de la classe JOptionPane són

Method Name	Description
showConfirmDialog	Asks a confirming question, like yes/no/cancel.
showInputDialog	Prompt for some input.
showMessageDialog	Tell the user about something that has happened.
showOptionDialog	The Grand Unification of the above three.

JOptionPane

- Capçalera d'alguns mètodes de la classe JOptionPane:

public static String **showInputDialog**(Object message) throws HeadlessException

public static String **showInputDialog**(Object message, Object initialSelectionValue)

public static String **showInputDialog**(Component parentComponent, Object message) throws HeadlessException

public static String **showInputDialog**(Component parentComponent, Object message, Object initialSelectionValue)

public static String **showInputDialog**(Component parentComponent, Object message, String title, int messageType) throws HeadlessException

public static Object **showInputDialog**(Component parentComponent, Object message, String title, int messageType, Icon icon, Object[] selectionValues, Object initialSelectionValue) throws HeadlessException

public static void **showMessageDialog**(Component parentComponent, Object message) throws HeadlessException

public static void **showMessageDialog**(Component parentComponent, Object message, String title, int messageType) throws HeadlessException

public static void **showMessageDialog**(Component parentComponent, Object message, String title, int messageType, Icon icon) throws HeadlessException

JOptionPane

- Capçalera d'alguns mètodes de la classe JOptionPane:

public static int **showConfirmDialog**(Component parentComponent, Object message) throws HeadlessException

public static int **showConfirmDialog**(Component parentComponent, Object message, String title, int optionType) throws HeadlessException

public static int **showConfirmDialog**(Component parentComponent, Object message, String title, int optionType, int messageType) throws HeadlessException

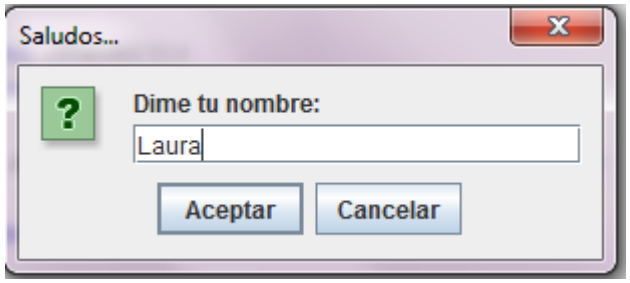
public static int **showConfirmDialog**(Component parentComponent, Object message, String title, int optionType, int messageType, Icon icon) throws HeadlessException

public static int **showOptionDialog**(Component parentComponent, Object message, String title, int optionType, int messageType, Icon icon, Object[] options, Object initialValue) throws HeadlessException

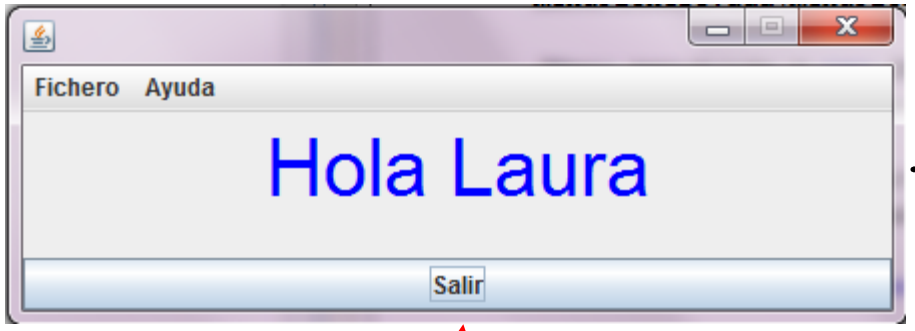
Exercici 2

- L'aplicació tindrà aquestes dues finestres:

Primera finestra:



Segona finestra:



Sortir de l'aplicació

Sortir de l'aplicació

S'obre una finestra per donar informació sobre qui ha fet l'aplicació

Comentaris

- Indica quins són els imports necessaris.

- Mètodes de la classe JOptionPane:

public static String **showInputDialog**(Component parentComponent, Object message, String title, int messageType) throws HeadlessException

- ERROR_MESSAGE
- INFORMATION_MESSAGE
- WARNING_MESSAGE
- QUESTION_MESSAGE
- PLAIN_MESSAGE

```
public class MenuDialogSwing2{
   private void go() {
      JFrame frame = new JFrame();
      frame.getContentPane().setLayout(new BorderLayout());
      frame.addWindowListener(.....);


      GestorMenus gestorMenus = new GestorMenus(frame);

      JMenu menuFichero = new JMenu("Fichero");
      JMenuItem menuFicheroSalir = new JMenuItem("Salir",'S');
      menuFicheroSalir.addActionListener(gestorMenus);
      menuFichero.add(menuFicheroSalir);

      JMenu menuAyuda = new JMenu("Ayuda");
      JMenuItem menuAyudaAcercaDe = new JMenuItem("Acerca de...",'A');
      menuAyudaAcercaDe.addActionListener(.....);
      menuAyuda.add(menuAyudaAcercaDe);

      JMenuBar barraMenu = new JMenuBar();
      barraMenu.add(menuFichero);
      barraMenu.add(menuAyuda);
      frame.getContentPane().add(barraMenu,BorderLayout.NORTH);
```

L'aplicació es tanca quan es tanca la finestra.

Objecte escoltador

?

```
public class MenuDialogSwing2{
   private void go() {
      JFrame frame = new JFrame();
      frame.getContentPane().setLayout(new BorderLayout());
      frame.addWindowListener(new WindowAdapter() {
                public void windowClosing(WindowEvent e){
                   System.exit(0);
                }
             });

      GestorMenus gestorMenus = new GestorMenus(frame);

      JMenu menuFichero = new JMenu("Fichero");
      JMenuItem menuFicheroSalir = new JMenuItem("Salir",'S');
      menuFicheroSalir.addActionListener(gestorMenus);
      menuFichero.add(menuFicheroSalir);

      JMenu menuAyuda = new JMenu("Ayuda");
      JMenuItem menuAyudaAcercaDe = new JMenuItem("Acerca de...",'A');
      menuAyudaAcercaDe.addActionListener(gestorMenus);
      menuAyuda.add(menuAyudaAcercaDe);

      JMenuBar barraMenu = new JMenuBar();
      barraMenu.add(menuFichero);
      barraMenu.add(menuAyuda);
      frame.getContentPane().add(barraMenu,BorderLayout.NORTH);
```

L'aplicació es tanca quan es tanca la finestra.

Objecte escoltador

Registem l'objecte menu a l'objecte escoltador

```
    JPanel panelNombre = new JPanel();
    panelNombre.setLayout(new FlowLayout());
    frame.getContentPane().add(panelNombre,BorderLayout.CENTER);
    JButton botonSalir = new JButton("Salir");
    frame.getContentPane().add(botonSalir,BorderLayout.SOUTH);
    botonSalir.addActionListener(.....);
```

?

```
    JLabel etiquetaNombre = new JLabel();
    panelNombre.add(etiquetaNombre);
    etiquetaNombre.setForeground(Color.blue);
    etiquetaNombre.setFont(new Font("Arial",Font.PLAIN,40));
    String nombre =.....;
    etiquetaNombre.setText("Hola " + nombre);
    frame.setBounds(250,200,440,150);
    frame.setVisible(true);
    frame.setResizable(false);
  }

  public static void main (String[] args) {
    MenuDialogSwing2 menu = new MenuDialogSwing2();
    menu.go();
  }
} // Final classe MenuDialogSwing2
```

?

Mètode main

```
        JPanel panelNombre = new JPanel();
        panelNombre.setLayout(new FlowLayout());
        frame.getContentPane().add(panelNombre,BorderLayout.CENTER);
        JButton botonSalir = new JButton("Salir");
        frame.getContentPane().add(botonSalir,BorderLayout.SOUTH);
        botonSalir.addActionListener(new ActionListener(){
                public void actionPerformed(ActionEvent evt) {
                   System.exit(0);
                }});
```

L'aplicació es tanca quan es prem el botó "Salir".

```
    JLabel etiquetaNombre = new JLabel();
        panelNombre.add(etiquetaNombre);
        etiquetaNombre.setForeground(Color.blue);
        etiquetaNombre.setFont(new Font("Arial",Font.PLAIN,40));
        String nombre = JOptionPane.showInputDialog(frame,"Dime tu nombre:","Saludos...",
JOptionPane.QUESTION_MESSAGE);
        etiquetaNombre.setText("Hola " + nombre);
        frame.setBounds(250,200,440,150);
        frame.setVisible(true);
        frame.setResizable(false);
    }
    public static void main (String[] args) {
      MenuDialogSwing2 menu = new MenuDialogSwing2();
      menu.go();
    }
} // Final classe MenuDialogSwing2
```

Obrim finestra per demanar el nom.

Mètode main

```
class GestorMenus implements ActionListener {
        ..............
}
```

Classe externa que implementa ActionListener

```
class GestorMenus implements ActionListener {

   JFrame finestra;
   GestorMenus(JFrame v) {
      finestra = v;
   }
   public void actionPerformed(ActionEvent evt) {
      if(evt.toString().indexOf("Salir") != -1)
         System.exit(0);
      else if(evt.toString().indexOf("Acerca de...") != -1)
         JOptionPane.showMessageDialog(finestra, "Saludos...\nAutor: Laura", "Acerca de...",
         JOptionPane.INFORMATION_MESSAGE);
   }
}
```

Classe externa que implementa ActionListener :
- per sortir si seleccionem l'opció "Salir" del menú "Fichero"
- per obrir una finestra amb informació de l'aplicació en cas de seleccionar l'opició "Acerca de…" del menú "Ayuda".

Comentaris

- En l'exemple, com que la classe GestorMenus és una classe externa, hem optat pel tipus de pregunta:

```
if(evt.toString().indexOf("Salir") != -1)
```

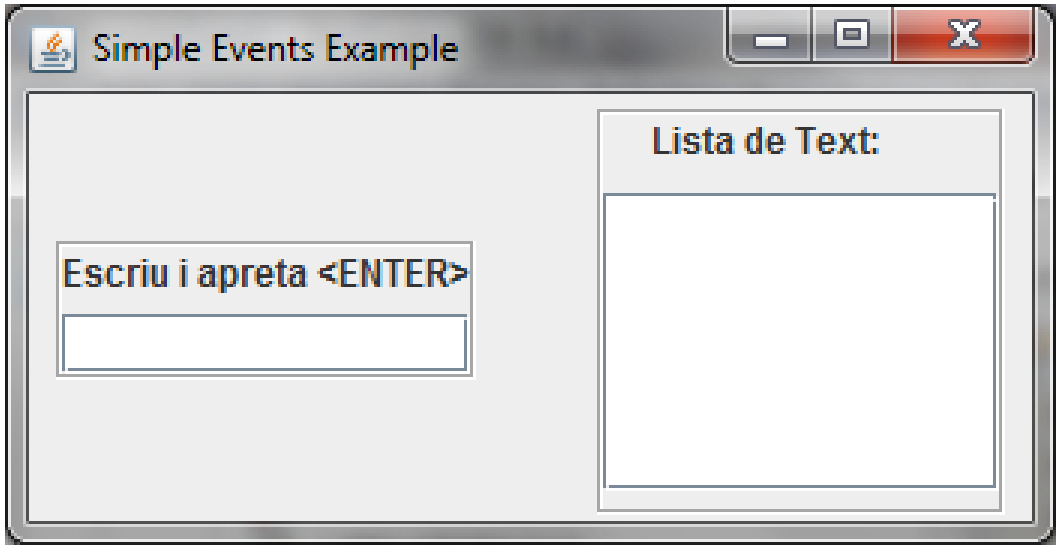
- En cas de tenir accés al menus, podriem haver utilitzat el mètode getSource() de ActionEvent.
- Aquest mètode torna l'objecte en el qual es produeix l'Event inicialment.

Comentaris

- Més mètodes de ActionEvent:
- String **getActionCommand**()
  Returns the command string associated with this action.
- int **getModifiers**()
  Returns the modifier keys held down during this action event.
- long **getWhen**()
  Returns the timestamp of when this event occurred.
- String **paramString**()
  Returns a parameter string identifying this action event.

Exemple 5: Panel d'escriptura

Implementem aquesta interfície on podem escriure text al camp de l'esquerra i quan premem enter apareixerà a la lista de text.



Exemple 5: implementació A

```
// importa els símbols de AWT i Swing
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
public class SimpleEventsA{
   // ample i alt del frame
   static final int WIDTH=350;
   static final int HEIGHT=180;

   // Declara JTextField per a entrar text
   JTextField textField;
   // Declara JTextArea per rebre línies de textField
   JTextArea textList;
   // Declara JScrollPane per a JTextArea
   JScrollPane pane;

   public static void main(String args[]) {
      SimpleEventsA gui = new SimpleEventsA();
      gui.go();
   }// Fi mètode main
```

Per definir el tamany de la finestra.

Atributs

```
// Mètode go: aquí es fa quasi tot el treball

  public void go(){
    /******* Crea un contenidor per a textField ****/
    // Instancia un JPanel
    JPanel textPanel = new JPanel();
    // li posa un borde (per defecte no en te)
    textPanel.setBorder(BorderFactory.createEtchedBorder());
    // Fixa el layout del textPanel a BorderLayout
    textPanel.setLayout(new BorderLayout());

    // Crea una etiqueta i la afegeix al panell
    JLabel textTitle =new JLabel("Escriu i apreta <ENTER>");
    textPanel.add(textTitle, BorderLayout.NORTH);

    // Instancia un JTextField i l'afegeix al panell
    textField = new JTextField();
    textPanel.add(textField, BorderLayout.SOUTH);

    // Afegeix un strut al textPanel com a marge inferior
    textPanel.add(Box.createVerticalStrut(6));
```

← JPanel

← Afegim JLabel

← Afegim JTextField

← Afegim Marge

```
/******** Crea un contenidor pel textArea ******/
// Instancia un JPanel
JPanel listPanel = new JPanel();
// afegeix borde
listPanel.setBorder (BorderFactory.createEtchedBorder());
// Set el layout del textPanel
listPanel.setLayout(new BoxLayout(listPanel,BoxLayout.Y_AXIS));
```

```
// Crea una etiqueta i afegeix al panel
JLabel title = new JLabel("Llista de Text:");
listPanel.add(title);
// Afegeix un strut al BoxLayout
listPanel.add(Box.createVerticalStrut(10));
// Instancia una JTextArea sense text inicial
// 6 files, 10 columnes, i vertical scrollbars
textList = new JTextArea("", 6, 10);
// la fem read-only (només de lectura)
textList.setEditable(false);
```

```
// Afegeix textList a listPanel
pane = new JScrollPane(textList);
listPanel.add(pane);
// Afegeix un strut a listPanel com a margen inferior
listPanel.add(Box.createVerticalStrut(6));
```

```
// Afegeix un listener a textField quan se pulsa ENTER copia el text de
//textField a l'area de text. Les components estan interrelacionades
textField.addActionListener(new CampText());
```

← Registrem escoltador

```
// Afegeix els 2 panels al frame, separats per strut
JFrame frame = new JFrame("Simple Events Example");
frame.setLayout (new FlowLayout());
frame.add(textPanel);
frame.add(Box.createHorizontalStrut(30));
frame.add(listPanel);
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame.setSize(WIDTH, HEIGHT);
frame.setVisible(true);
```

}//Fi mètode go

```
class CampText implements ActionListener{
    public void actionPerformed(ActionEvent e) {
        // Afegeix el text de textField a textList
        textList.append(textField.getText());
        textList.append("\n");
        // Reset el textField
        textField.setText("");
    }
}
```

Classe interna que implementa la interfície ActionListener

```
}// Fi de la classe SimpleEvent
```

Exemple 5: implementació B

```
// importa els símbols de AWT and Swing
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class SimpleEventsB extends JFrame {

   // ample i alt del frame
   static final int WIDTH=350;
   static final int HEIGHT=180;
   // Declara JTextField per a entrar text
   JTextField textField;
   // Declara JTextArea per rebre línies de textField
   JTextArea textList;
   // Declara JScrollPane per a JTextArea
   JScrollPane pane;
```

```
public static void main(String args[]) {
    SimpleEventsB frame = new SimpleEventsB("Simple Events Example");
    // Standard adapter usat en quasi totes les
    // aplicacions per a tancar la finestra
    frame.addWindowListener(new WindowAdapter() {
       @Override
       public void windowClosing(WindowEvent e) {
       System.exit(0);
       }
    });
    // fixa el tamany de frame i el mostra
    frame.setSize(WIDTH, HEIGHT);
    frame.setVisible(true);
  }// Fi mètode main
```

Mètode main

```
// Constructor: aquí es fa quasi tot el treball
public SimpleEventsB(String lab) {
   // crida al constructor de JFrame: posa etiqueta
   super(lab);

   /******* Crea un contenidor per a textField ****/
   // Instancia un JPanel
   JPanel textPanel = new JPanel();
   // li posa un borde (per defecte no en te)
   textPanel.setBorder(BorderFactory.createEtchedBorder());
   // Fixa el layout del textPanel a BorderLayout
   textPanel.setLayout(new BorderLayout());
   // Crea una etiqueta i la afegeix al panel
   JLabel textTitle =new JLabel("Escriu i apreta <ENTER>");
   textPanel.add(textTitle, BorderLayout.NORTH);
   // Instancia un JTextField i afegeix a textPanel
   textField = new JTextField();
   textPanel.add(textField, BorderLayout.SOUTH);
   // Afegeix un strut al textPanel com a marge inferior
    textPanel.add(Box.createVerticalStrut(6));
```

```
/******** Crea un contenidor pel textArea ******/
// Instancia un JPanel
JPanel listPanel = new JPanel();
// afegeix borde
listPanel.setBorder (BorderFactory.createEtchedBorder());
// Set el layout del textPanel
listPanel.setLayout(new BoxLayout(listPanel,BoxLayout.Y_AXIS));
// Crea una etiqueta i afegeix al panel
JLabel title = new JLabel("Lista de Text:");
listPanel.add(title);
// Afegeix un strut al BoxLayout
listPanel.add(Box.createVerticalStrut(10));
// Instancia una JTextArea sense text inicial
// 6 files, 10 columnes, i vertical scrollbars
textList = new JTextArea("", 6, 10);
// la fem read-only (només de lectura)
textList.setEditable(false);
// Afegeix textList a listPanel
pane = new JScrollPane(textList);
listPanel.add(pane);
// Afegeix un strut a listPanel com a margen inferior
listPanel.add(Box.createVerticalStrut(6));
```

```
    // Afegeix un listener a textField quan se pulsa ENTER copia el text de
    //textField a l'area de text. Les components estan interrelacionades
    textField.addActionListener(new ActionListener() {
       public void actionPerformed(ActionEvent e) {
          // Afegeix el text de textField a textList
          textList.append(textField.getText());
          textList.append("\n");
          // Reset el textField
          textField.setText("");
       }});

    // Afegeix els 2 panels al frame, separats per strut
    Container c = getContentPane();
    c.setLayout (new FlowLayout());
    c.add(textPanel);
    c.add(Box.createHorizontalStrut(30));
    c.add(listPanel);
  }//Fi mètode constructor


}// Fi de la classe SimpleEvent
```

Classe interna anònima que implementa la interfície ActionListener

LOOK AND FEEL

Look and feel

- Swing està dissenyat per a que es pugui canviar el *look and feel* d'una aplicació GUI
  - ***look*** fa referència a l'**aparença** de les components de la GUI
  - ***feel*** fa referència al **funcionament** de les components de la GUI

- Els programes Java poden adoptar l'aparença de la plataforma sobre la que s'executen, o aparences especifiques.

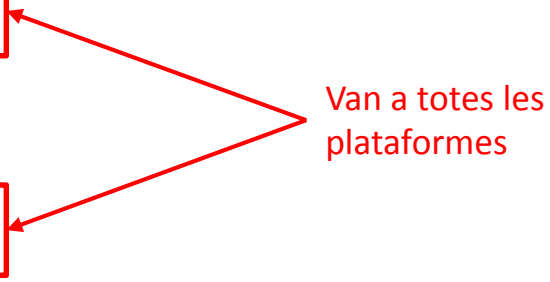
Look and feel

- *CrossPlatformLookAndFeel* or *Metal*

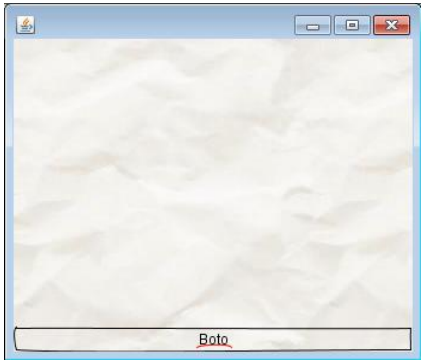
  Standard Java look and feel. Forma part de l'API (javax.swing.plaf.metal) i s'utilitza per defecte.
- *SystemLookAndFeel*: look and feel natiu per la plataforma. Dins del Java SDK.

Look and feel

- Java proporciona:
  - Metal
  - Nimbus
  - Motif
  - Windows
  - Windows classic



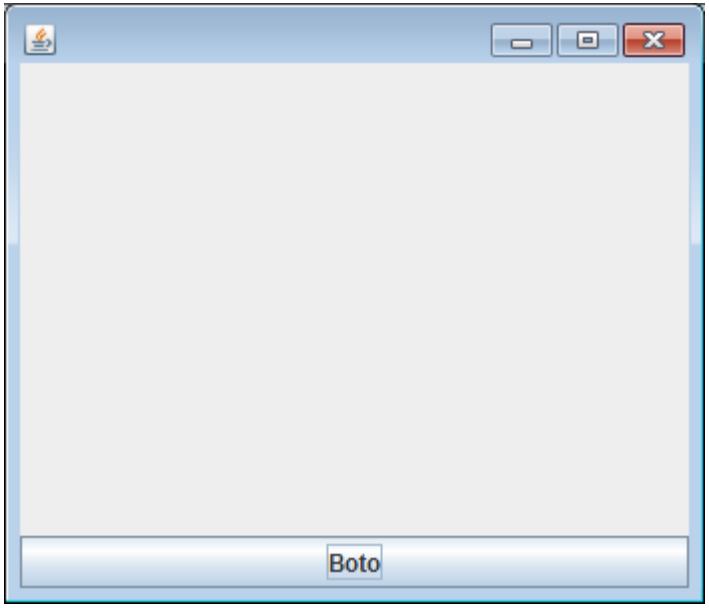
Look and feel




```
import napkin.NapkinLookAndFeel;
public class ExempleLookFeel{
    public static void main(String [] args){
        ExempleLookFeel ex = new ExempleLookFeel();
       ex.goLookFeelNapkin();
     }
      public void goLookFeelNapkin(){
       try {
           UIManager.setLookAndFeel(new NapkinLookAndFeel());
       } catch (Exception unused) {
         // Ignorar l'excepció, ja que no podem fer res.  Utilitzarem el default.
       }
       JFrame frame = new JFrame();
       JButton button = new JButton("Boto");
       frame.getContentPane().add(BorderLayout.SOUTH, button);
       frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
       frame.setSize(350,300);
       frame.setVisible(true);
    }
}
```

Creem una instància d'aquesta classe LookAndFeel

Look and feel



Default Look And Fell

NapkinLookAndFeel

Exemple Colors

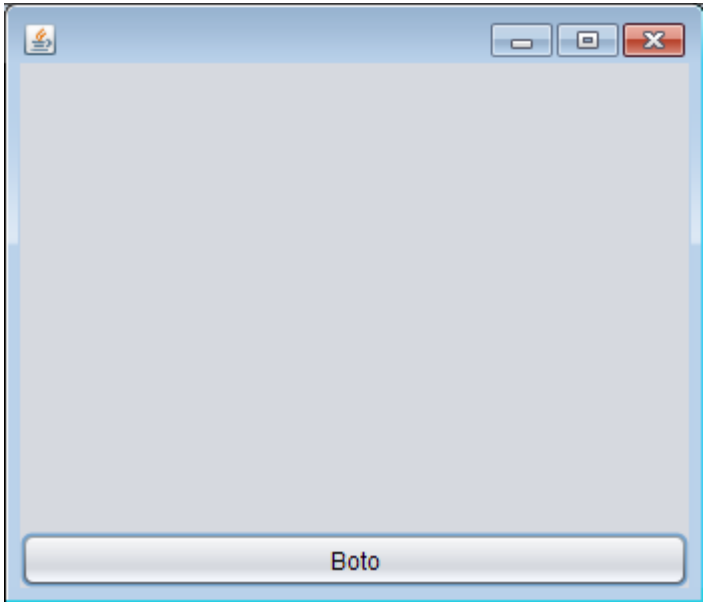

```
public class ExempleLookFeelProves {

   public static void main(String [] args){
      ExempleLookFeelProves exemple = new ExempleLookFeelProves();
      exemple.go();
   }

   public void go(){
            JFrame frame = new JFrame();
            JButton button = new JButton("Boto");
            button.setBackground(Color.red);
            frame.getContentPane().add(BorderLayout.SOUTH, button);

            frame.getContentPane().setBackground(Color.green);
            frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
            frame.setSize(350,300);
            frame.setVisible(true);
    }
}
```

Canvia el color de fons del JButton

Canvia el color de fons del JFrame

Look and feel

Color Look And Fell



nimbus Look And Feel

Look and feel

- Per fixar el look-and-feel podem utilitzar:

```
UIManager.LookAndFeelInfo plafinfo[] = UIManager.getInstalledLookAndFeels();
 boolean nimbusfound = false;
int nimbusindex = 0;
for (int look = 0; look < plafinfo.length; look++) {
    if (plafinfo[look].getClassName().toLowerCase().contains("nimbus")) {
          nimbusfound = true;
          nimbusindex = look;
    }
}
try {
    if (nimbusfound) {
       UIManager.setLookAndFeel(plafinfo[nimbusindex].getClassName());
    } else {
       UIManager.setLookAndFeel(UIManager.getCrossPlatformLookAndFeelClassName());
    }
} catch (Exception e) {       }
```

PANELLS I GRÀFICS

Panells i Gràfics

- Hem treballat amb panells utilitzant-los com a contenidors d'altres components. D'aquesta forma podíem establir el diagrama de la interfície definint el diagrama de cada panell.
- Un panell es pot utilitzar també per a dibuixar imatges
- Veurem llavors cóm utilitzar un objecte de la classe JPanel com un àrea dedicada específicament al dibuix.
- L'usuari podrà dibuixar amb el ratolí o podrà dibuixar determinats objectes gràfics predeterminats de les llibreries gràfiques.

Graphics

1. Posar components en una finestra:

   frame.getContentPane().add(myBoto);

2. Dibuixar un gràfic sobre un component

   Graphics.fillOval(70,70,100,100);

3. Posar un JPEG en una component:

   Graphics.grawImage(myPic,10,10,this);

Exemple 6: GUI simple

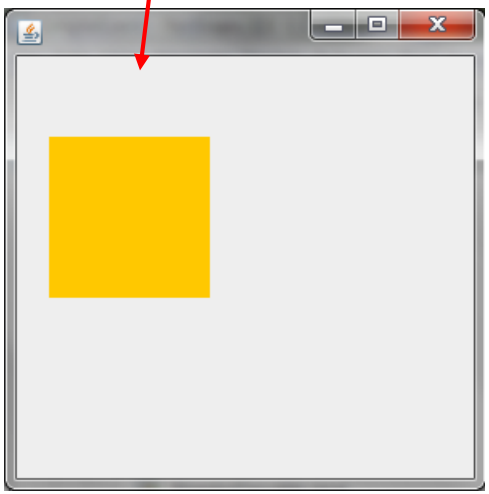
```
import java.awt.Color;
import java.awt.Graphics;
import javax.swing.JFrame;
import javax.swing.JPanel;

public class GUIPanelDeDibuix {
  public static void main (String[] args) {
     GUIPanelDeDibuix gui = new GUIPanelDeDibuix();
     gui.go();
  }

  public void go() {
     JFrame frame = new JFrame();
      frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
     PanelDeDibuix drawPanel = new PanelDeDibuix();
     frame.getContentPane().add(drawPanel);
     frame.setSize(300,300);
     frame.setVisible(true);
  }// tanca mètode go()
}// tanca classe GUIPanelDeDibuix
```

```
class PanelDeDibuix extends JPanel {

   public void paintComponent(Graphics g) {
      g.setColor(Color.orange);
      g.fillRect(20,50,100,100);
   }
}
```

Amplada i alçada.

Posició cantonada esquerra dalt

paintComponent



Mètode paintComponent()

- Mai cridar al mètode paintComponent() directament.

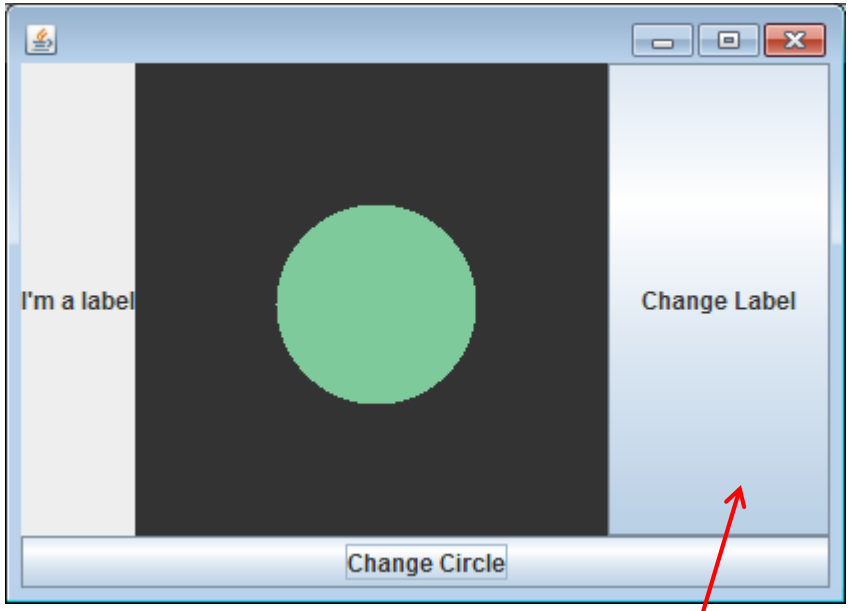
- Qui crida **paintComponent()?**
  - Es crida automàticament quan es fa visible la component.
  - Es crida indirectament des del listener definit per l'usuari via repaint()
    - Canvi de les variables de la instància.
    - Crida a repaint().

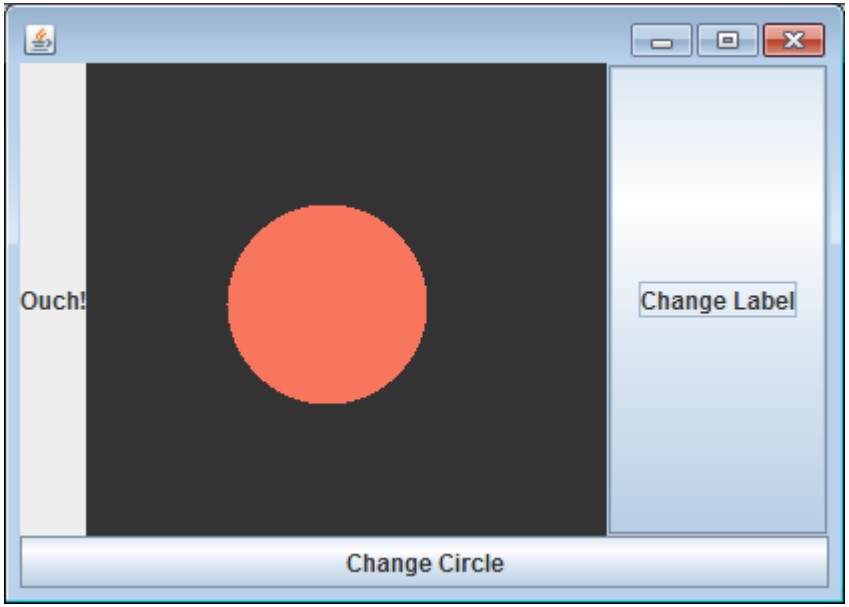
Component

- Mètode repaint()
- public void repaint() -> pinta tota la component
- public void repaint(long tm)
- public void repaint(int x, int y, int width, int height)
- public void repaint(long tm, int x, int y, int width, int height)

Exemple 7

- Implementeu una interfície gràfica d'usuari que contingui dos botons. Amb un canviem el color del cercle i amb l'altre canviem l'etiqueta





Aquest botó canvia el color del cercle

Aquest botó canvia el text de l'altra part de la finestra

Exemple 7

Classes internes

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
public class TwoButtons  {
   JFrame frame;
   JLabel label;
   public static void main (String[] args) {
      TwoButtons gui = new TwoButtons();
      gui.go();
   }
   public void go() {
      frame = new JFrame();
      frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

      JButton labelButton = new JButton("Change Label");
      labelButton.addActionListener(new LabelButtonListener());

      JButton colorButton = new JButton("Change Circle");
      colorButton.addActionListener(new ColorButtonListener());

      label = new JLabel("I'm a label");
      PanelDeDibuix drawPanel = new PanelDeDibuix ();
      frame.getContentPane().add(BorderLayout.SOUTH, colorButton);
      frame.getContentPane().add(BorderLayout.CENTER, drawPanel);
      frame.getContentPane().add(BorderLayout.EAST, labelButton);
      frame.getContentPane().add(BorderLayout.WEST, label);

      frame.setSize(420,300);
      frame.setVisible(true);
   }
```

Mètode main

Crida al mètode paintComponent

```
   class LabelButtonListener implements ActionListener{
      public void actionPerformed(ActionEvent event) {
         label.setText("Ouch!");
      }
   }
   class ColorButtonListener implements ActionListener{
      public void actionPerformed(ActionEvent event) {
         frame.repaint();
      }
   }
} // Fi class TwoButtons
```

Exemple 7

…. Classes internes

```
class LabelButtonListener implements ActionListener{
  public void actionPerformed(ActionEvent event) {
     label.setText("Ouch!");
  }
}
class ColorButtonListener implements ActionListener{
  public void actionPerformed(ActionEvent event) {
     frame.repaint();
  }
}
} // Fi class TwoButtons
```

Crida al mètode paintComponent

paintComponent és un mètode de JComponent

Refresca el panell de dibuix

```
class PanelDeDibuix extends JPanel {
 @Override
   public void paintComponent(Graphics g) {
      g.fillRect(0,0,this.getWidth(), this.getHeight());
      // make random colors to fill with
      int red = (int) (Math.random() * 255);
      int green = (int) (Math.random() * 255);
      int blue = (int) (Math.random() * 255);

      Color randomColor = new Color(red, green, blue);
      g.setColor(randomColor);
      g.fillOval(70,70,100,100);
    }
} // Fi class PanelDeDibuix
```

Els següents gràfics es pintaran amb el color aleatori definit

EXEMPLES D’ANIMACIONS

Exemple 8: Animació simple

- Implementeu una interfície gràfica que dibuixa sobre una finestra de tamany 300x300 pixels un cercle verd a la posició inicial (70,70) i el va movent en diagonal fins arribar a la cantonada inferior dreta.

Exemple 8: Animació simple

```
import javax.swing.*;
import java.awt.*;
public class SimpleAnimation {
  int x = 70;
  int y = 70;
  public static void main (String[] args) {
    SimpleAnimation gui = new SimpleAnimation ();
    gui.go();
  }
  public void go() {
    JFrame frame = new JFrame();
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    PanelDeDibuix drawPanel = new PanelDeDibuix();
    frame.getContentPane().add(drawPanel);
    frame.setSize(300,300);
    frame.setVisible(true);
    for (int i = 0; i < 130; i++) {
      x++;
      y++;
      drawPanel.repaint();
      try {
        Thread.sleep(5);
      } catch(Exception ex) { }
    }
  }// close go() method
```

Mètode main

L'acció està aquí:

Es repeteix 130 vegades

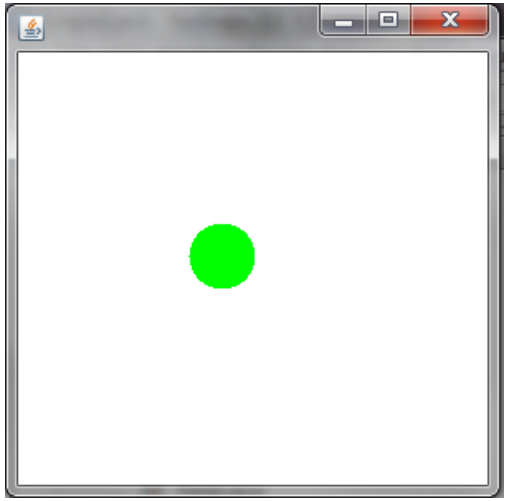
Li diu al panel que es torni a pintar

Espera 5 milisegons

```
class PanelDeDibuix extends JPanel {
  @Override
  public void paintComponent(Graphics g) {
    g.setColor(Color.white);
    g.fillRect(0,0,this.getWidth(), this.getHeight());

    g.setColor(Color.green);
    g.fillOval(x,y,40,40);

  }
 } // close inner class
} // close outer class
```

El blanc és el primer color seleccionat

El verd és el segon



Exemple 9: Animació simple

- Implementeu una interfície gràfica que dins d'una finestra de tamany 500x300 pixels dibuixa un rectangle blau d'amplada 500 i alçada 250 i el va fent més petit fins que desapareix.

Exemple 9: Una altra animació

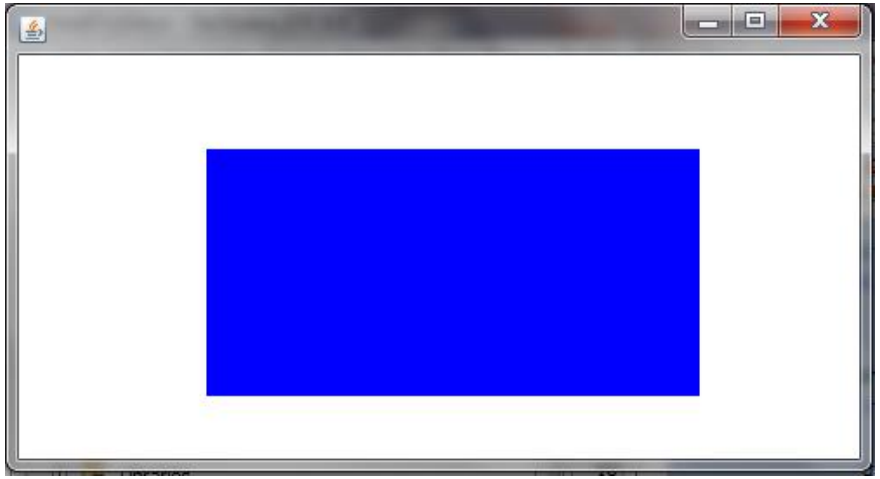
```
import javax.swing.*;
import java.awt.*;
public class Animate {
   private static final int XSIZE = 500;
   private static final int YSIZE = 300;
   int x = 1;
   int y = 1;
```

Mètode main

```
   public static void main (String[] args) {
      Animate gui = new Animate ();
      gui.go();
   }
   public void go() {
      JFrame frame = new JFrame();
      frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      PanelDeDibuix panelDibuix = new PanelDeDibuix();
      frame.getContentPane().add(panelDibuix);
      frame.setSize(XSIZE,YSIZE);
      frame.setVisible(true);
      for (int i = 0; i < 125; i++) {
         panelDibuix.repaint();
         try {
           Thread.sleep(50);
         } catch(Exception ex) { }
         x=x+2;
         y++;
      }
   }// close go() method
```

El rectangle és dues vegades més ample que alt

```
   class PanelDeDibuix extends JPanel {
         public void paintComponent(Graphics g ) {
               g.setColor(Color.white);
               g.fillRect(0,0,XSIZE,YSIZE);
               System.out.println("x,y" + x +" , " +y);
               g.setColor(Color.blue);
               g.fillRect(x,y,500-x*2,250-y*2);
         }
      }// Fi classe PanelDeDibuix
   } // Fi classe Animate
```



Exemple 10: animació amb el ratolí

- Implementeu una interfície gràfica de tamany 400x200 pixels on al apretar amb el ratolí sobre ella apareixen al costat els valors de les coordenades de la posició del ratolí en aquest moment.

RatonSwing Animation

```
public class RatonSwing{
   int x,y;
   JFrame frame;
   PanelDeDibujo panelDeDibujo;
```

Mètode main

```
   public static void main (String[] args) {
      RatonSwing rat = new RatonSwing();
      rat.go();
   }

   private void go(){
      frame = new JFrame("Jugando con el ratón ...");
      frame.addWindowListener(new GestorFinestra());

      panelDeDibujo = new PanelDeDibujo();
      frame.getContentPane().add(panelDeDibujo,BorderLayout.CENTER);
      frame.addMouseListener(new GestorRatoli());

      frame.setBounds(250,200,400,200);
      frame.setVisible(true);
   }
```

Continua en la següent pàgina….

```
    class GestorRatoli extends MouseAdapter{
        @Override
        public void mousePressed(MouseEvent e) {
            x = e.getX();
            y = e.getY();
            frame.repaint();
        }
    }
```

Classe interna que hereta de la classe MouseAdapter i gestiona l'event de prémer el ratolí reescrivint el mètode mousePressed.

Crida al mètode paint de la classe PanelDeDibujo.

```
    class PanelDeDibujo extends JPanel {
         PanelDeDibujo(){
            setBackground(Color.white);
         }
        public void paint(Graphics g) {
            g.drawString(x+","+y,x,y);
        }
    }
} Final classe RatonSwing
```

Classe interna que hereta de Jpanel i reescriu el mètode paint

```
class GestorFinestra extends WindowAdapter{
    @Override
    public void windowClosing(WindowEvent e) {
        System.exit(0);
    }
}
```

Classe que hereta de WindowAdapter i reescriu el mètode windowClosing.

Referències del bloc 3

- Llibre "**Head First Java**", Kathy Sierra & Bert Bates.
- *Apunts: "***Aprenda Java como si estuviera en Primero**" (Universidad de Navarra): http://www.tecnun.es/asignaturas/Informat1/ayudainf/aprendainf/Java/Java2.pdf
- "**Creating a GUI with JFC/Swing**" (The Swing Tutorial) http://java.sun.com/docs/books/tutorial/uiswing/
- The Swing Connection http://java.sun.com/javase/technologies/desktop/articles.jsp
- Components: http://download.oracle.com/javase/tutorial/uiswing/components/