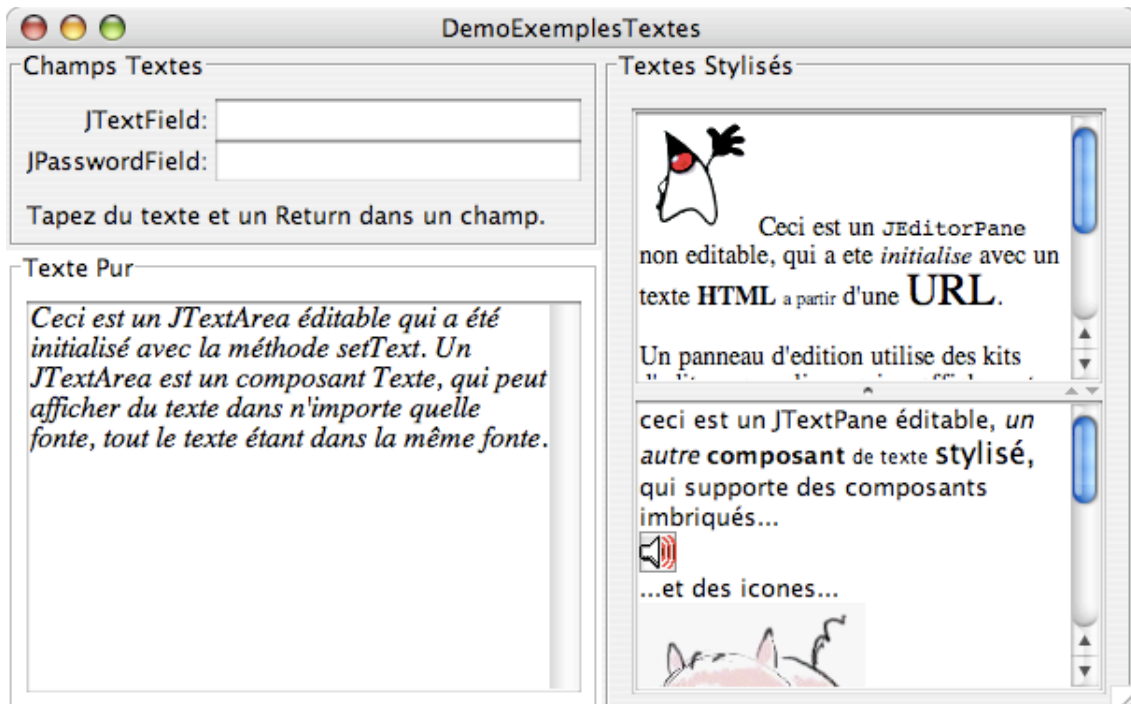


Feuille 7

On va tester les différentes sortes d'éditeurs de texte de Swing. La fenêtre principale aura à la fin de ce long TP l'allure suivante :



On va implémenter l'agencement des différentes zones de cette fenêtre dans une architecture globale, mais on va procéder pas à pas dans le sujet, pour implanter et tester d'abord séparément les différents composants de cette fenêtre.

Architecture globale : le panneau de contenu du `JFrame` sera constitué de 2 panneaux disposés dans une `Box` horizontale. Le panneau de gauche comportera lui-même deux panneaux dans une `Box` verticale : un panneau avec un bord titré « Champs Textes » qui sera géré par un `GridBagLayout`, et un panneau avec un bord titré « Texte Pur » qui comportera un `JScrollPane` encadrant une `JTextArea`. A droite, le panneau de droite contiendra un `JSplitPane` réunissant un `JEditorPane` (dans un `JScrollPane`) et un `JTextPane` (dans un `JScrollPane` également).

Exercice 1 : Panneau titré « Champs Textes ». Commencer par créer un `JFrame` affichant ce panneau dans sa zone centrale.

Pour l'alignement des titres des champs et des champs textes, on pourra utiliser une méthode

```
private void addLabelTextRows(JLabel[] labels,  
                              JTextField[] textFields,  
                              GridBagConstraints gridbag,  
                              Container container) { ...}
```

prenant en argument un tableau de `JLabels` et un tableau de `JTextField` de même taille ainsi que le `GridBagConstraints` qui servira de gestionnaire d'affichage au dernier argument de type conteneur destiné à contenir les labels et les textfields. Cette méthode aligne les labels devant les textfields à l'aide du `GridBagConstraints` et les ajoute au conteneur.

Mais libre à vous de procéder différemment en agencant simplement deux champs textes.

Votre `JFrame` implémentera l'écouteur `ActionListener` pour définir des actions sur les champs textes. Le `JLabel` situé en dessous (sur l'image « Tapez du texte et un return dans un champs ») pourra par exemple afficher un autre message lors de l'entrée de caractères dans les champs étiquetés `JTextField` et `JPasswordField` (messages du genre « Vous avez tapé xxxx »).

Exercice 2 : Créer maintenant le panneau complet de gauche de la figure. Pour la `JTextArea`, on utilisera une fonte « Serif » de taille 16 et en italique.

Exercice 3 : Créer maintenant le `JFrame` complet. Dans le `JSplitPane` situé à droite, le `JEditorPane` sera non éditable (contrairement aux autres éditeurs de la fenêtre) et initialisé à l'aide d'une URL avec la méthode `setPage`. Prévoir la possibilité d'une erreur de lecture de la page au moyen d'une exception. Pour le `JTextPane`, vous pourrez vous amuser à y mettre quelques icônes, ou à changer de fonte par exemple

Correction:

```
import javax.swing.*;  
import javax.swing.text.*;  
  
import java.awt.*;           //for layout managers  
import java.awt.event.*;    //for action and window events  
  
import java.net.URL;  
import java.io.IOException;  
  
public class TextSamplerDemoFr extends JFrame  
    implements ActionListener {  
    private String newline = "\n";  
    protected static final String textFieldString = "JTextField";
```

```

protected static final String passwordFieldString = "JPasswordField";

protected JLabel actionLabel;

public TextSamplerDemoFr() {
    super("DemoExemplesTextes");

    //Create a regular text field.
    JTextField textField = new JTextField(10);
    textField.setActionCommand(textFieldString);
    textField.addActionListener(this);

    //Create a password field.
    JPasswordField passwordField = new JPasswordField(10);
    passwordField.setActionCommand(passwordFieldString);
    passwordField.addActionListener(this);

    //Create some labels for the fields.
    JLabel textFieldLabel = new JLabel(textFieldString + ": ");
    textFieldLabel.setLabelFor(textField);
    JLabel passwordFieldLabel = new JLabel(passwordFieldString + ": ");
    passwordFieldLabel.setLabelFor(passwordField);

    //Create a label to put messages during an action event.
    actionLabel = new JLabel("Tapez du texte et un Return dans un
champ.");
    actionLabel.setBorder(BorderFactory.createEmptyBorder(10,0,0,0));

    //Lay out the text controls and the labels in a textControlsPane.
    JPanel textControlsPane = new JPanel();
    GridBagLayout gridbag = new GridBagLayout();
    GridBagConstraints c = new GridBagConstraints();

    textControlsPane.setLayout(gridbag);

    JLabel[] labels = {textFieldLabel, passwordFieldLabel};
    JTextField[] textFields = {textField, passwordField};
    // rajoute les textFields a droite en face des labels a gauche
    addLabelTextRows(labels, textFields, gridbag, textControlsPane);

    c.gridwidth = GridBagConstraints.REMAINDER; //last
    c.anchor = GridBagConstraints.WEST;
    c.weightx = 1.0;
    gridbag.setConstraints(actionLabel, c);
    textControlsPane.add(actionLabel);
    textControlsPane.setBorder(
        BorderFactory.createCompoundBorder(
            BorderFactory.createTitledBorder("Champs Textes"),
            BorderFactory.createEmptyBorder(5,5,5,5)));

    //Create a text area.
    JTextArea textArea = new JTextArea(

```

```

        "Ceci est un JTextArea éditable " +
        "qui a été initialisé avec la méthode setText. " +
        "Un JTextArea est un composant Texte, " +
        "qui peut afficher du texte dans n'importe " +
        "quelle fonte, tout le texte étant dans la même fonte."
    );
    textArea.setFont(new Font("Serif", Font.ITALIC, 16));
    textArea.setLineWrap(true);
    textArea.setWrapStyleWord(true);
    // une areaScrollPane pour l'empaqueter
    JScrollPane areaScrollPane = new JScrollPane(textArea);
    areaScrollPane.setVerticalScrollBarPolicy(
        JScrollPane.VERTICAL_SCROLLBAR_ALWAYS);
    areaScrollPane.setPreferredSize(new Dimension(250, 250));
    areaScrollPane.setBorder(
        BorderFactory.createCompoundBorder(
            BorderFactory.createCompoundBorder(
                BorderFactory.createTitledBorder("Texte Pur"),
                BorderFactory.createEmptyBorder(5,5,5,5)),
            areaScrollPane.getBorder()));

    //Create an editor pane.
    JEditorPane editorPane = createEditorPane();
    JScrollPane editorScrollPane = new JScrollPane(editorPane);
    editorScrollPane.setVerticalScrollBarPolicy(
        JScrollPane.VERTICAL_SCROLLBAR_ALWAYS);
    editorScrollPane.setPreferredSize(new Dimension(250, 145));
    editorScrollPane.setMinimumSize(new Dimension(10, 10));

    //Create a text pane.
    JTextPane textPane = createTextPane();
    JScrollPane paneScrollPane = new JScrollPane(textPane);
    paneScrollPane.setVerticalScrollBarPolicy(
        JScrollPane.VERTICAL_SCROLLBAR_ALWAYS);
    paneScrollPane.setPreferredSize(new Dimension(250, 155));
    paneScrollPane.setMinimumSize(new Dimension(10, 10));

    //Put the editor pane and the text pane in a split pane.
    JSplitPane splitPane = new JSplitPane(JSplitPane.VERTICAL_SPLIT,
        editorScrollPane,
        paneScrollPane);
    splitPane.setOneTouchExpandable(true);
    JPanel rightPane = new JPanel();
    rightPane.add(splitPane);
    rightPane.setBorder(BorderFactory.createCompoundBorder(
        BorderFactory.createTitledBorder("Textes
Stylisés"),
        BorderFactory.createEmptyBorder(5,5,5,5)));

    //Put everything in the applet.
    JPanel leftPane = new JPanel();

```

```

BoxLayout leftBox = new BoxLayout(leftPane, BoxLayout.Y_AXIS);
leftPane.setLayout(leftBox);
leftPane.add(textControlsPane);
leftPane.add(areaScrollPane);

JPanel contentPane = new JPanel();
BoxLayout box = new BoxLayout(contentPane, BoxLayout.X_AXIS);
contentPane.setLayout(box);
contentPane.add(leftPane);
contentPane.add(rightPane);
setContentPane(contentPane);
}

private void addLabelTextRows(JLabel[] labels,
                              JTextField[] textFields,
                              GridBagLayout gridbag,
                              Container container) {
    GridBagConstraints c = new GridBagConstraints();
    c.anchor = GridBagConstraints.EAST; // partie visible si rétrécit
    int numLabels = labels.length;

    for (int i = 0; i < numLabels; i++) {
        c.gridwidth = GridBagConstraints.RELATIVE; //next-to-last
        c.fill = GridBagConstraints.NONE; //reset to default
        c.weightx = 0.0; //reset to default
        gridbag.setConstraints(labels[i], c);
        container.add(labels[i]);

        c.gridwidth = GridBagConstraints.REMAINDER; //end row
        c.fill = GridBagConstraints.HORIZONTAL;
        c.weightx = 1.0; // (absorbe l'espace en x)
        gridbag.setConstraints(textFields[i], c);
        container.add(textFields[i]);
    }
}

public void actionPerformed(ActionEvent e) {
    String prefix = "Vous avez tapé \";
    if (e.getActionCommand().equals(textFieldString)) {
        JTextField source = (JTextField)e.getSource();
        actionLabel.setText(prefix + source.getText() + "\");
    } else {
        JPasswordField source = (JPasswordField)e.getSource();
        actionLabel.setText(prefix + new String(source.getPassword())
                            + "\");
    }
}

private JEditorPane createEditorPane() {
    JEditorPane editorPane = new JEditorPane();
    editorPane.setEditable(false);
    String s = null;
}

```

```

try {
    s = "file:"
        + System.getProperty("user.dir")
        + System.getProperty("file.separator")
        + "TextSamplerDemoHelpFr.html";
    URL helpURL = new URL(s);
    displayURL(helpURL, editorPane);
} catch (Exception e) {
    System.err.println("N'ai pas pu créer l'URL help: " + s);
}
return editorPane;
}

private void displayURL(URL url, JEditorPane editorPane) {
    try {
        editorPane.setPage(url);
    } catch (IOException e) {
        System.err.println("Essai de lire une mauvaise URL: " + url);
    }
}

private JTextPane createTextPane() {
    JTextPane textPane = new JTextPane();
    String[] initString =
        { "ceci est un JTextPane éditable, ", //regular
          "un autre ", //italic
          "composant ", //bold
          "de texte ", //small
          "stylisé, ", //large
          "qui supporte des composants imbriqués..." +
newline, //regular
          " " + newline, //button
          "...et des icônes..." + newline, //regular
          " ", //icon
          newline + "JTextPane est une sous-classe de JEditorPane
qui " + "utilise un StyledEditorKit et un StyledDocument, et fournit " +
          "des méthodes pour interagir avec ces objets."
        };

    String[] initStyles =
        { "regular", "italic", "bold", "small", "large",
          "regular", "button", "regular", "icon",
          "regular"
        };

    initStylesForTextPane(textPane);

    Document doc = textPane.getDocument();

    try {
        for (int i=0; i < initString.length; i++) {
            doc.insertString(doc.getLength(), initString[i],

```

```

        textPane.getStyle(initStyles[i]));
    }
} catch (BadLocationException ble) {
    System.err.println("N'ai pas pu inserer le texte initial.");
}

return textPane;
}

protected void initStylesForTextPane(JTextPane textPane) {
    //Initialize some styles.
    Style def = StyleContext.getDefaultStyleContext().
        getStyle(StyleContext.DEFAULT_STYLE);

    Style regular = textPane.addStyle("regular", def);
    StyleConstants.setFontFamily(def, "SansSerif");

    Style s = textPane.addStyle("italic", regular);
    StyleConstants.setItalic(s, true);

    s = textPane.addStyle("bold", regular);
    StyleConstants.setBold(s, true);

    s = textPane.addStyle("small", regular);
    StyleConstants.setFontSize(s, 10);

    s = textPane.addStyle("large", regular);
    StyleConstants.setFontSize(s, 16);

    s = textPane.addStyle("icon", regular);
    StyleConstants.setAlignment(s, StyleConstants.ALIGN_CENTER);
    StyleConstants.setIcon(s, new ImageIcon("images/Pig.gif"));

    s = textPane.addStyle("button", regular);
    StyleConstants.setAlignment(s, StyleConstants.ALIGN_CENTER);
    JButton button = new JButton(new ImageIcon("images/sound.gif"));
    button.setMargin(new Insets(0,0,0,0));
    button.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            Toolkit.getDefaultToolkit().beep();
        }
    });
    StyleConstants.setComponent(s, button);
}

public static void main(String[] args) {
    JFrame frame = new TextSamplerDemoFr();

    frame.addWindowListener(new WindowAdapter() {
        public void windowClosing(WindowEvent e) {
            System.exit(0);
        }
    });
}

```

```
    });  
    frame.pack();  
    frame.setVisible(true);  
  }  
}
```

fichier TextSamplerDemoHelpFr.html

```
<html>  
<body>  
  
Ceci est un <code>JEditorPane</code> non editable,  
qui a ete <em>initialise</em>  
avec un texte <strong>HTML</strong> <font size=-2>a partir</font> d'une  
<font size=+2">URL</font>.  
<p>  
Un panneau d'edition utilise des kits d'editeur  
pour lire, ecrire, afficher, et editer du texte dans  
differeents formats.  
Le package de texte Swing inclut des kits d'edition  
pour le plein texte, HTML, et RTF.  
Vous pouvez aussi developper  
un editeur prevu pour d'autres formats.  
</body>  
</html>
```