Jean-François Berdjugin, Pierre-Alain Jacquot, Vincent Lestideau
IUT 1, département SRC, l'Isle d'Abeau.

M2.23.5 Programmation

Durée : 2h45
Documents autorisés.
Le barème est donné à titre indicatif.
A la fin du TP vous devrez rendre une archive de projet sur ftp://ftp-exam.src.

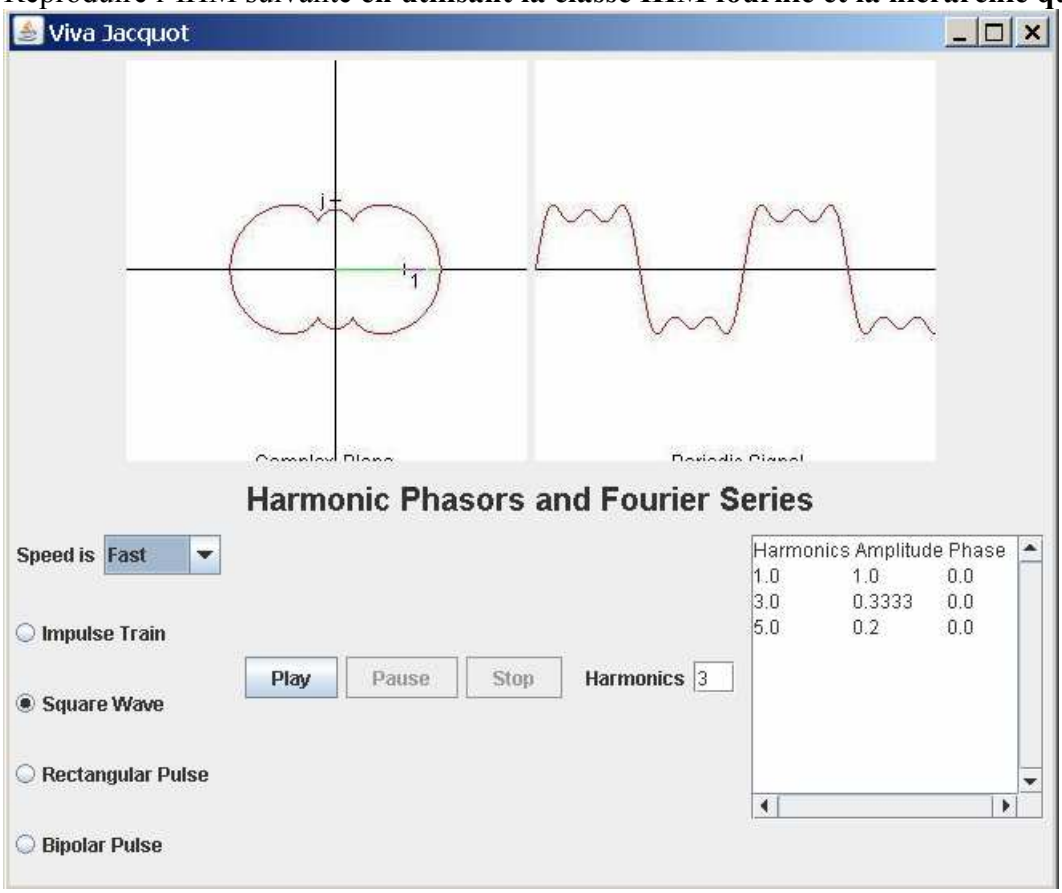
Préambule

Créer sur le disque **public** un répertoire portant votre **nom**. Lancer *eclipse* et choisir le répertoire **précédent** comme **Workspace**. Importer (« *existing projects into workspace* ») dans ce workspace l'archive disponible sur **http://serveur.pedago.src/~administrateur**.

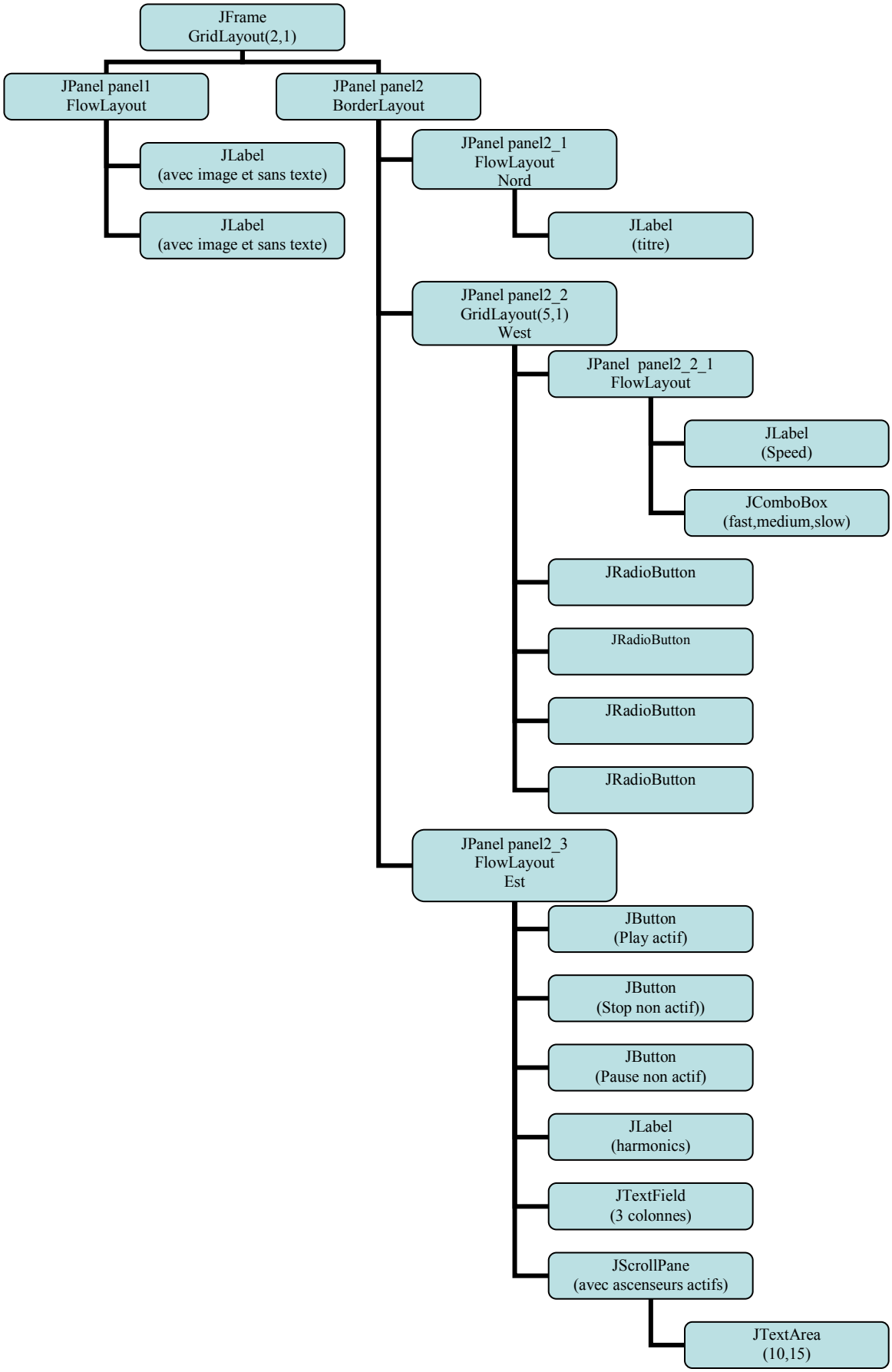
Cette archive contient un paquetage nommé ds_machine qui contient les classes utiles pour les événements et le paquetage ds_machine.ressources qui contient les images pour l'IHM. A la fin vous exporterez le **projet après l'avoir renommé** (click-droit->refactor->rename) avec votre nom et prénom dans une archive portant votre nom et vous prénom que vous déposerez sur le serveur ftp.

IHM (10 points)

Reproduire l'IHM suivante **en utilisant la classe IHM fournie et la hiérarchie qui suit**.



Hiérarchie :


JFrame
GridLayout(2,1)
JPanel panel1
FlowLayout
JPanel panel2
BorderLayout
JLabel
(avec image et sans texte)
JLabel
(avec image et sans texte)
JPanel panel2_1
FlowLayout
Nord
JLabel
(titre)
JPanel panel2_2
GridLayout(5,1)
West
JPanel panel2_2_1
FlowLayout
JLabel
(Speed)
JComboBox
(fast,medium,slow)
JRadioButton
JRadioButton
JRadioButton
JRadioButton
JPanel panel2_3
FlowLayout
Est
JButton
(Play actif)
JButton
(Stop non actif))
JButton
(Pause non actif)
JLabel
(harmonics)
JTextField
(3 colonnes)
JScrollPane
(avec ascenseurs actifs)
JTextArea
(10,15)

Comportement

Je vous conseil de dessiner le tout avant de fixer un comportement, mais le comportement suivant doit être reproduit :

- Seul un JRadioButton peut être actif à la fois, initialement c'est « square wave » ; pour grouper des JRadioButton, il faut un ButtonGroup.
- Le boutton « play » est actif, « pause » et « stop » ne le sont pas.
- Le JTextField doit-être initialisé à « 3 »
- La JTextArea doit être remplie
- La JComboBox doit afficher « fast »

Recommandations

Les recommandations suivantes sont faites pour vous aider.

JLabel avec image

- Pour mettre une image sur un JLabel, il faut créer une ImageIcon et l'associer au JLabel, l'url est relative au projet. Ici vous utiliserez, par exemple : "./bin/ds_machine/ressources/image2.jpg". Le JLabel doit être créé sans texte.

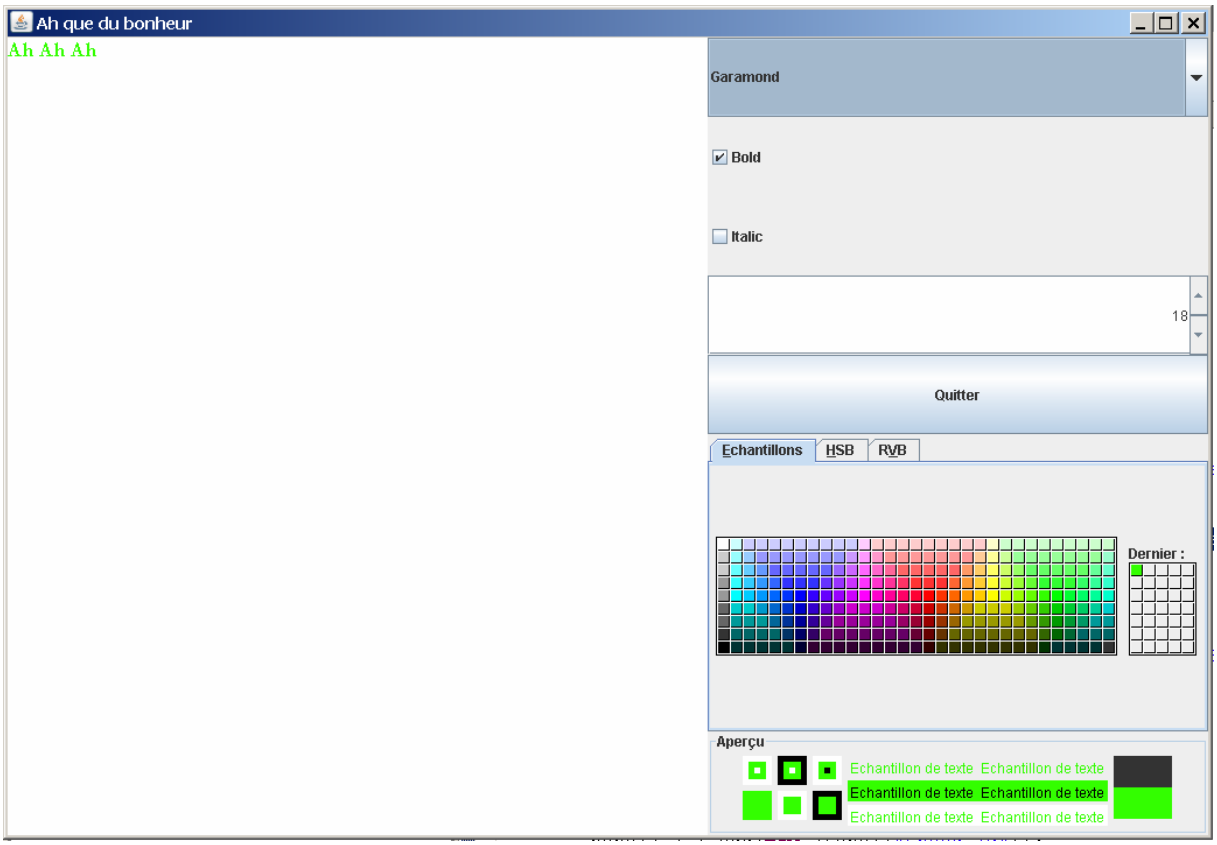
ButtonGroup

- Les ButtonGroup permettent de regrouper des boutons. Un ButtonGroup n'est pas un composant, il n'est pas nécessaire de l'afficher par contre il doit être peuplé avec les JRadioButton

Tabulation et retour à la ligne

- Le caractère *\n* permet un retour à la ligne
- Le caractère *\t* permet un saut de tabulation

Evénements (10 points)



Compléter le code de la classe *Evenements* pour ajouter le comportement suivant, ce comportement permet de modifier l'aspect de la textArea (`textArea`) à la suite d'une interaction avec l'un des éléments suivants :

- Le boutton (`buttonExit`) permet de quitter l'application.
- Le spinner (`spinnerSize`) permet de modifier la taille de la police.
- La comboBox (`comboBoxFont`) permet de modifier la police.
- Les chekBox (*checkBoxItalic, checkBoxBold*) permettent de modifier le style
- Le JColorChooser (*colorChooser*) permet de choisir la couleur.

Les modifications de la JTextArea doit être faite en appelant la méthode public *void updateTextArea()*. Cette méthode vous est fournie vide à vous de la compléter et de l'appeler dans vos classes « écouteurs ».
Les classes d'écouteur devront être implémentées sous forme de **classes internes**.

Rem :

- Vous aurez besoin d'au moins deux gestionnaires d'événement.
- Hors mis pour quitter les traitements sont les mêmes.
- L'écouteur de s'applique pas sur le JColorChooser mais sur son *ColorSelectionModele* (*colorChooser.getSelectionModel().addXXXListener(YYY)*)