

Développement Logiciel, 2013

TP5 Interfaces graphiques

Très important : Ce TP est noté. Vous devez envoyer par e-mail les sources du TP dans une archive à mounir.assaf@cea.fr avant le **dimanche 03 Mars 2013 à 23h59**. Il s'agit d'un travail personnel à effectuer en monôme ; les binômes sont donc interdits.

On se propose de créer une interface graphique pour le gestionnaire de série implémenté lors du TP3. Nous vous proposons pour cela une correction du TP3 sur laquelle vous baserez vos développements. Vous pouvez récupérer une archive contenant le sujet à cette adresse : <http://db.tt/gZEcbLIi>. **Attention, il vous est interdit de modifier les classes fournis.**

La fenêtre principale *GestionnaireUI* de notre application hérite d'une *JFrame*. Cette fenêtre permet la création, le chargement et la sauvegarde d'un gestionnaire grâce à un menu.

Cette fenêtre permet aussi l'affichage de la liste de séries du gestionnaire, et l'ajout d'une série au gestionnaire.

Un double-click sur une série du gestionnaire déclenche l'ouverture d'une nouvelle fenêtre *SerieUI* héritant de *JDialog*. Cette dernière fenêtre permet d'afficher une table d'épisodes de la série (numéro d'épisode, le titre, la date de diffusion, vu/pas vu?). Elle permet aussi de filtrer les épisodes vus.

Nous utiliserons principalement l'API Swing¹ lors de nos développements.

Question 1: Nous allons commencer par créer une classe *Gestionnaire.gui.GestionnaireUI* héritant de *JFrame*.

1. Implémentez une méthode *init()* qui sera appelée par le constructeur de *GestionnaireUI* pour initialiser la *JFrame* (titre, taille, position centrée au lancement, visibilité).
2. Implémentez une méthode *initMenu()* qui sera appelée par le constructeur. Cette méthode crée un menu nommé 'Gestionnaire' avec les sous-menus suivant :
 - Nouveau : permet la création d'un nouveau gestionnaire. Il remplace tout simplement l'ancien gestionnaire par un nouveau.
 - Charger : permet le chargement d'un gestionnaire existant. Demande à l'utilisateur de choisir un fichier de type '*.save' grâce au composant *JFileChooser* avant de le charger grâce à la méthode *EntreesSorties.chargerFichier(File f)*. Un fichier de test 'gestionnaire.save' vous est fourni. Contentez-vous pour l'instant d'afficher le résultat d'appel de la méthode *toString()* sur ce gestionnaire pour tester le chargement. Attention toutefois, si vous modifiez sensiblement les fichiers sources fournis, la désérialisation du fichier 'gestionnaire.save' fourni ne marchera plus (*InvalidClassException local class incompatible...*).
 - Sauvegarder : permet la sauvegarde du gestionnaire courant. Il demande à l'utilisateur de choisir un fichier pour sauvegarder le gestionnaire courant (*EntreesSorties.sauvegarderFichier(Gestionnaire g, File f)*). Veillez toutefois à surcharger la méthode *public void approveSelection()* de *JFileChooser* afin de demander une confirmation à l'utilisateur dans le cas où il sélectionne un fichier existant, afin de ne pas écraser le fichier sélectionné par mégarde. Une méthode simple pour demander une confirmation consiste à utiliser la méthode *public static int JOptionPane.showConfirmDialog(...)*.

1. Vous trouverez des tutoriels très complets pour les composants Swing à cette adresse : <http://docs.oracle.com/javase/tutorial/uiswing/components/index.html>

```

1 int result = JOptionPane.showConfirmDialog(this, "The file exists,
    overwrite?", "Existing file", JOptionPane.YES_NO_CANCEL_OPTION);
    if (result == JOptionPane.YES_OPTION){ ... }
3 else{ ... }

```

Question 2: Nous nous occuperons maintenant de l’affichage des séries du gestionnaire. Pour cela, nous utilisons une *JList*. Afin de pouvoir mettre à jour le contenu de cette *JList*, nous utilisons une instance de *DefaultListModel*. Tous les changements de cette instance, sont répercutés sur la *JList* automatiquement.

1. Implémentez une méthode *initListSeries()* appelée par le constructeur. Cette méthode crée une *JList* puis la rajoute au contenu de *GestionnaireUI*. Pensez au cas où la liste des séries serait trop grandes (*JScrollPane*).
2. Implémentez ensuite la méthode *initSeriesIntoList()* qui réinitialise le contenu de la *JList* avant d’y ajouter toutes les séries du gestionnaire. À ce stade, vous devriez pouvoir charger le fichier ‘gestionnaire.save’ fourni et voir les différentes séries s’afficher.



FIGURE 1 – GestionnaireUI

3. Rajouter maintenant un bouton ‘ajouter série’ permettant de charger une série depuis un fichier ‘.csv’ et de l’ajouter au gestionnaire. Vous pouvez utiliser un *BoxLayout* pour forcer l’alignement vertical de la *JList* et du bouton ‘ajouter série’. La figure 1 illustre un exemple du résultat attendu. L’utilisateur devrait pouvoir fournir le nom de la série, puis le fichier à charger. Vous pouvez pour cela créer un *JDialog* demandant à l’utilisateur ces informations. Notez que si vous déclarez ce *JDialog* en tant qu’*Inner Class*, vous pouvez accéder simplement aux champs de *GestionnaireUI*. Notez aussi que la modification seule de l’instance de *DefaultListModel* ne suffit pas à mettre à jour le gestionnaire. . .

Pour vous faciliter la tâche et vous permettre de vous familiariser avec les *Layout managers*, nous vous proposons de créer un premier conteneur *JPanel* dont le *layout manager* est un *GridLayout(2,2)* (deux lignes, deux colonnes). Ce conteneur *JPanel* contient un *JLabel* "nom série", un *JTextField*, un *JLabel* "Fichier :)" et enfin un *JButton* déclenchant l'ouverture d'un *JFileChooser*. Ce conteneur *JPanel* ainsi qu'un bouton *Valider* sont ensuite alignés sur le conteneur principal du *JDialog* en utilisant un *BoxLayout* vertical. La figure 2 illustre un exemple du résultat attendu.

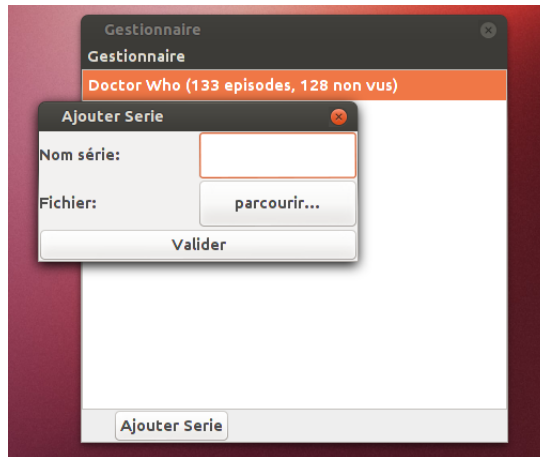


FIGURE 2 – Ajouter une série dans GestionnaireUI

4. Vérifiez que l'utilisateur n'a pas rentré un nom de série vide, et a bien choisi un fichier. Sinon, affichez un message d'erreur (*JOptionPane.showMessageDialog*).
5. Ajouter un Listener qui permet de déclencher l'affichage du résultat de l'appel de *toString()* sur une instance de série lorsque l'utilisateur double-clique sur cette série.
6. Rajouter un petit logo (un fichier 'monster.png' vous est fourni) au dessus de la *JList*. Pour cela, créez une classe héritant de *JComponent* puis veillez à surcharger la méthode *paintComponent(Graphics g)* afin de dessiner l'image. Veillez à ce que cette image reste centrée dans le cas où la fenêtre est redimensionnée.

Question 3: Nous allons maintenant passer à l'affichage des épisodes d'une série. Pour cela, nous implémenterons une classe *SerieUI* héritant de *JDialog*, affichant une *JTable* d'épisodes. Cette fenêtre est ouverte lorsque l'utilisateur double-clique sur une série dans *GestionnaireUI*.

1. Créez la classe *Gestionnaire.gui.SerieUI* ainsi qu'une méthode *init()* appelé depuis le constructeur pour initialiser la fenêtre.
2. Implémentez une méthode *initListEpisodes()* appelé depuis le constructeur permettant de créer une *JTable* et de l'ajouter au contenu de la fenêtre. Puis implémentez une méthode *initEpisodesIntoTable()* pour initialiser les données de la *JTable* en y ajoutant les informations concernant les épisodes de la série (4 colonnes : episode, titre, diffusion, vu ?).

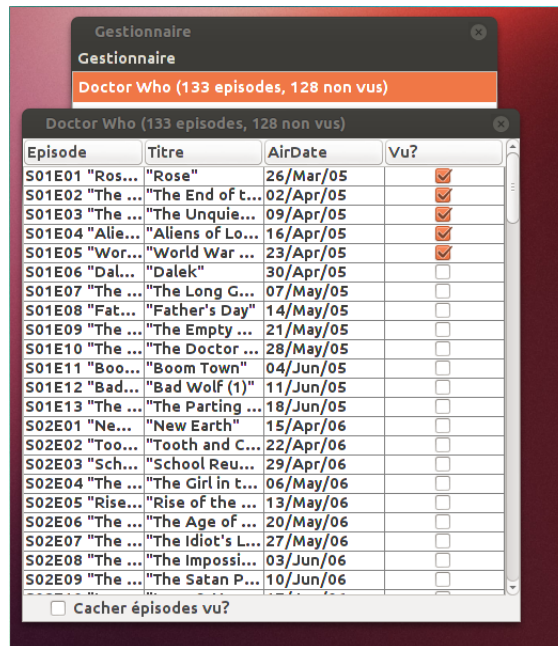


FIGURE 3 – SerieUI

3. Faites en sorte que la dernière colonne s'affiche comme une checkbox et qu'elle soit la seule colonne modifiable dans la JTable. La figure 3 illustre un exemple du résultat attendu.
4. Faites en sorte de mettre à jour la serie lorsque la valeur de la colonne vu? est mise à jour. Vous pouvez utiliser pour cela un *TableModelListener*.
5. Ajouter un Checkbox à la fenêtre SerieUI permettant de cacher tous les épisodes de la série déjà vu (*TableRowSorter* et *setRowFilter*).