

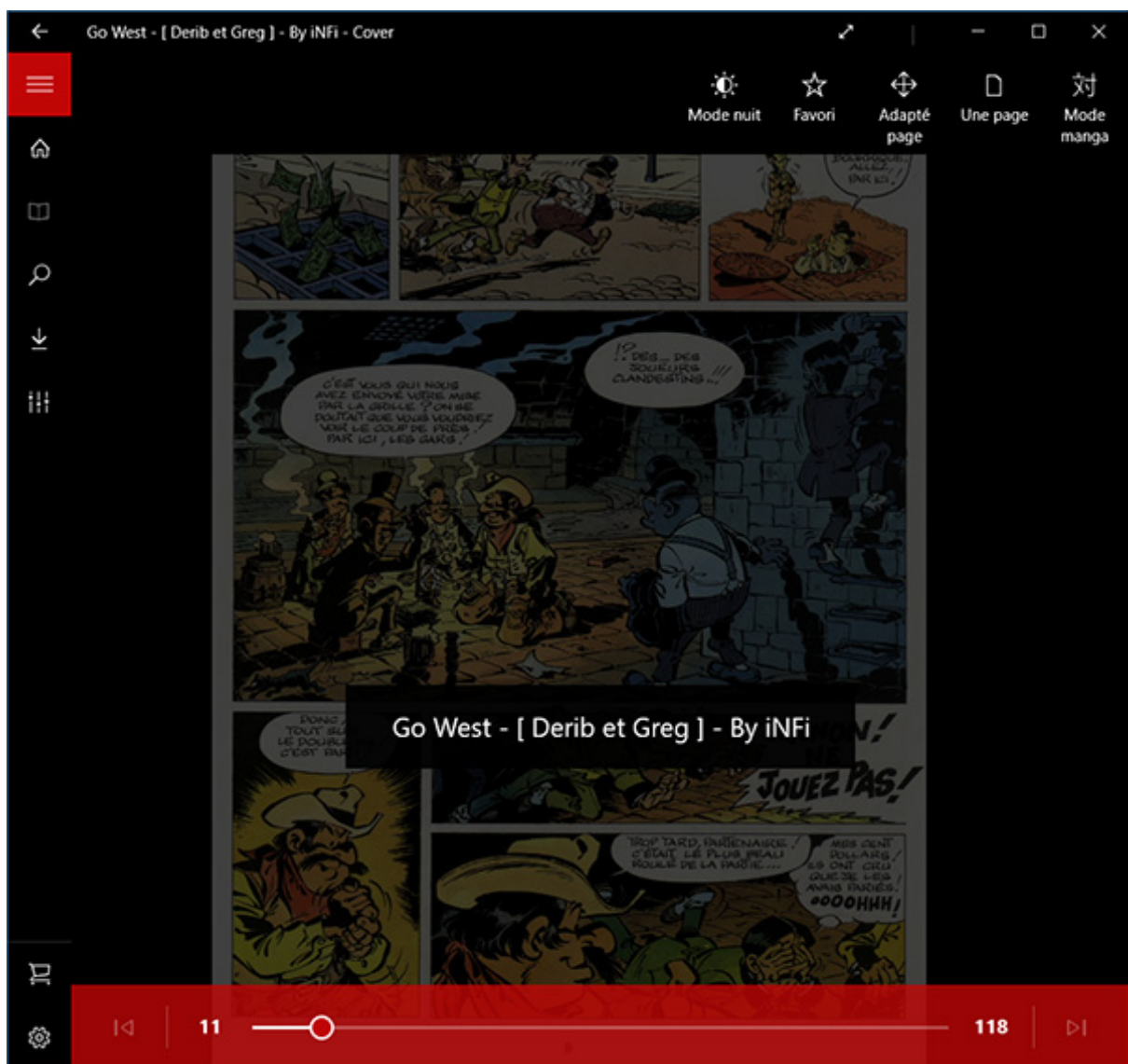
Lucas POLLET

Promo 2020/2021

Jean PERRUT

Info 2 Tp4

Rapport Projet Java : Lecture de BD



Exemple de lecteur de BD. Ceci n'est pas le projet.

Sommaire

I. Introduction

Présentation du projet

II. Analyse

- 1) Classes utilisés
- 2) Relations entre les classes
- 3) Fonctionnement global

III. Réalisation

- 1) Choix techniques
- 2) Présentation des algorithmes complexes utilisés

IV. Utilisation

- 1) Mode d'emploi
- 2) Configuration requise

V. Conclusion

- 1) Bilan
- 2) Optimisations possibles
- 3) Extensions possibles

I. Introduction

Présentation du projet :

Le projet consiste à créer un logiciel permettant d'afficher des Bandes Dessinées numériques dans une fenêtre, au même titre qu'une liseuse de livres électronique.

Pour être lue, la BD doit être au préalable au format CBZ (Comic Book archive Zipped) n'étant autre que des fichiers compressés avec ZIP contenant les pages de la BD sous forme d'images au format JPEG.

A cela s'ajoute la possibilité de zoomer et dézoomer les images si nécessaire ainsi que la reprise de lecture à la dernière page consultée avant la fermeture de l'application.

II. Analyse

1) Classes utilisées :

- Classe **Main** :
 - Classe principale de l'application. C'est à partir de cette classe que la fenêtre est instanciée et porte le nom de « Lecteur de BD ». Elle est intégrée au package « fr.lecteurbd ».
- Classe **BD** :
 - Classe permettant de créer une liste des pages de la BD. Ses méthodes permettent de retourner un élément de la liste voire une copie de la liste entière si besoin et le nom de la BD. Elle est intégrée au package « fr.lecteurbd.stock ».
- Classe **Page** :
 - Classe permettant de stocker les informations pour une page de la BD au moyen de ses méthodes telles que le numéro, le nom ou le chemin d'accès de la page. Elle est intégrée au package « fr.lecteurbd.stock ».
- Classe **OpenSelector** :
 - Classe permettant de rechercher et sélectionner la BD au format .cbz ou .zip dans un répertoire de la machine où elle est stockée. Elle est intégrée au package « fr.lecteurbd.utils ».
- Classe **UnzipFile** :
 - Classe permettant de décompresser la bande dessinée au format .cbz ou .zip, mais aussi de l'afficher image par image. Elle est intégrée au package « fr.lecteurbd.utils ».
- Classe **Home_Panel** :
 - Classe permettant d'initialiser les paramètres de la fenêtre de menu ainsi que les items à afficher à l'intérieur de celle-ci. Elle hérite de la Classe **JPanel**, implémente l'interface **ActionListener** et est intégrée au package « fr.lecteurbd.windows ».
- Classe **HomeWindow** :
 - Classe permettant d'instancier le contenu de la fenêtre de menu et la fenêtre en elle-même. Elle hérite de la classe **JFrame** et est intégrée au package « fr.lecteurbd.windows ».

- Classe **LectureWindow** :

- Classe permettant d'initialiser la fenêtre de défilement des images de la BD (le lecteur de BD entre autres) ainsi que sa barre d'outils. Elle hérite de la classe **Jframe**, implémente l'interface **ActionListener** et est intégrée au package « fr.lecteurbd.windows »

- Classe **Page_Panel** :

- Classe permettant de gérer le contenu du lecteur de BD (boutons précédent, suivant, image de fond) ainsi que la position de l'image visionnée par rapport à la fenêtre, le zoom et dé-zoom et les fonctions additionnelles au moyen de touche de clavier. Elle hérite de la classe **JPanel**, implémente les interfaces **ActionListener**, **MouseWheelListener**, **KeyListener** et est intégrée au package « fr.lecteurbd.windows »

2) Relations entre les classes :

- La classe **Main** fait appel à la classe **BD** et **HomeWindow** pour créer l'interface de l'application ainsi que la liste d'image à afficher.
- La classe **BD** utilise la classe **Page** en type de vecteur pour créer la liste d'élément de la BD.
- La classe **OpenSelector** fait appel à la classe **UnzipFile** afin que lorsqu'une BD au format zippé soit sélectionnée, elle suive l'étape de la décompression par la suite.
- La classe **UnzipFile** fait appel à la classe **LectureWindow** afin de lancer la lecture de la BD après décompression.
- La classe **Home_Panel** fait appel à la classe **OpenSelector** pour la sélection de fichier dans un répertoire de la machine.
- La classe **HomeWindow** fait appel à la classe **Home_Panel** pour ajouter le contenu de la fenêtre de menu.
- La classe **LectureWindow** fait appel à la classe **Page_Panel** pour ajouter le contenu de la fenêtre de lecture d'image de la BD.

3) Fonctionnement global :

A partir de la classe **Main** (classe principale), une instanciation de la classe **HomeWindow** est créée.

Il se trouve que la classe **HomeWindow** possède les caractéristiques de la classe **Home_panel** à travers une instanciation.

Dans **Home_panel** se trouve une méthode de la classe **OpenSelector** permettant de choisir un fichier compressé dans un répertoire de la machine.

Dans **OpenSelector** se trouve méthode de la classe **UnzipFile** permettant de décompresser la BD sélectionnée.

Dans **UnzipFile** se trouve une instanciation de la classe **LectureWindow** permettant d'afficher une fenêtre réglée automatiquement en plein écran.

Dans **LectureWindow** se trouve une instanciation de la classe **Page_panel** permettant la gestion de l'affichage de la BD ainsi que certains boutons dans cette même fenêtre.

Tout ce cheminement permet au bon fonctionnement de l'application et à l'utilisation de toutes les classes créées.

III. Réalisation

1) Choix techniques :

- Utilisation de la bibliothèque Swing pour l'interface graphique et la sélection de fichier depuis un répertoire de la machine.
- Création et utilisation d'un nouveau package nommé « fr.lecteurbd » dont les classes sont classés selon leur utilisation.
- Utilisation de la bibliothèque Awt par moment pour les évènements ainsi que pour l'interface graphique.
- Utilisation de la bibliothèque io pour gérer les exceptions et javax.ImageIO pour charger des images depuis un répertoire.

2) Présentation des algorithmes complexes :

Il n'y a pas vraiment d'algorithmes complexes pour ce projet, si ce n'est peut-être la gestion du zoom libre étant le produit d'une variable flottante prédéfinis et la rotation de la roulette de zoom puis l'actualisation de l'image avec repaint() :

```
zoom = Math.max(0, zoom - 0.03f * e.getWheelRotation());  
repaint();
```

3) Javadoc

Dans le dossier **doc/** , une documentation, générée à partir du code, permet d'en savoir plus sur chaque fonctions et variables du programme.

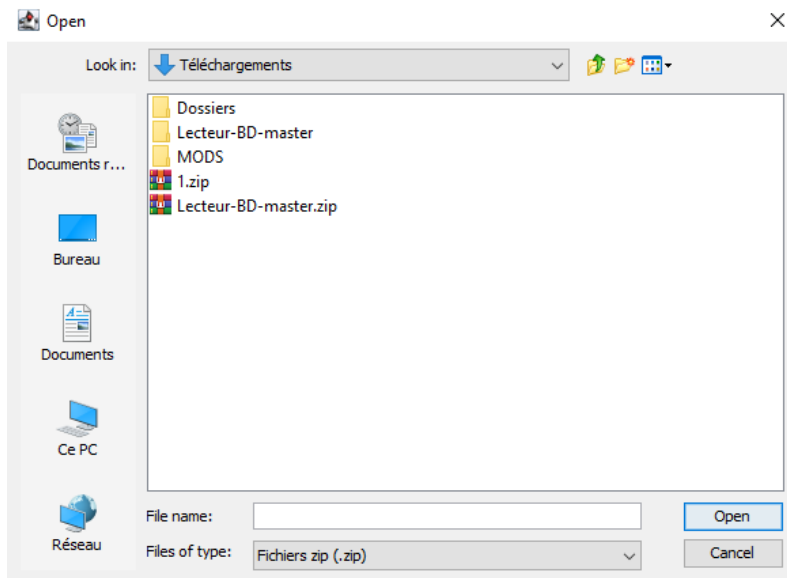
IV. Utilisation

1) Mode d'emploi :

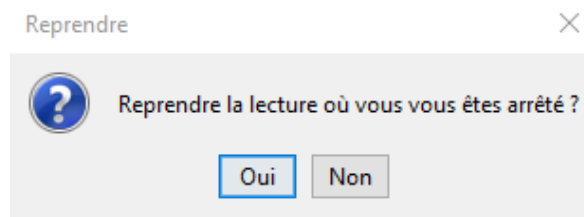
L'application fonctionne au travers de 2 principales fenêtres. La première correspond à la fenêtre de menu :



Elle possède un bouton ayant pour fonction d'ouvrir une BD depuis le répertoire de la machine. Lorsqu'on clique sur le bouton, une fenêtre apparaît permettant de choisir un fichier depuis la machine, comme ceci :



On sélectionne l'archive .zip ou .cbz correspondant à la BD voulue puis on clique sur open. Avant que la 2^{ème} fenêtre ne s'ouvre et n'affiche la BD, une boîte de dialogue apparaît permettant la reprise de lecture si une BD avait commencé à être lue.



Oui si on souhaite continuer là où l'on s'est arrêté, Non si on souhaite reprendre depuis le début.

Après cela, la 2^{ème} fenêtre apparaît et on peut d'ores et déjà commencer la lecture, selon le choix de reprise si la BD a été ouverte auparavant ou depuis le début si c'est une nouvelle BD à lire. Le changement de page se fait à l'aide des boutons précédent et suivant.



2) Configuration requise :

- Système : Un ordinateur sous Windows 32 ou 64 bits ayant une version de java récente d'installé (Version 8 à la date du 20 Octobre 2020)
- Résolution : Résolution minimale de 800x800 pixels
- Stockage : 30 Mo d'espace disponible

V. Conclusion

1) Bilan :

L'application est fonctionnelle car il remplit les objectifs telles qu'afficher dans une fenêtre des bandes dessinées numériques en chargeant des fichiers au format .cbz ou .zip, naviguer facilement de page en page grâce à des boutons, faire en sorte que l'image occupe toute la largeur de la fenêtre, utiliser un niveau de zoom libre et enfin afficher une BD à partir de la dernière page lue si on recharge une BD qui avait commencé à être lue.

2) Optimisations possibles :

- Un mode clair / sombre pour le confort des yeux de l'utilisateur selon ses préférences

- Une lecture automatique qui change de page toute les *n secondes (*n réglable selon le choix de l'utilisateur, 30 par défaut).

3) Extensions possibles :

Une extension possible de l'application serait de ne pas se limiter qu'au Bande dessinées mais à pouvoir l'utiliser pour voir/revoir des albums photos, ou encore lire un livre, un journal de presse, page par page.