Rapport de projet
Fiche récapitulative des contributions

Projet 2D_library

https://github.com/chunhu16/2D_library
Modèle : MVC

Contributions :

- SUQUET Olympie
  Remarque : Le contributeur Etienne c'est moi.
  - Création des fonctions de hachage et des fonctions equals pour les classes principales : Library, Book, Shelf, Author, et tests unitaires sur ces fonctions.
  - Initialisation du modèle : création des classes Library, Author, Book et Shelf.
  - Reprise du code de génération d'image pour générer le fond de la bibliothèque en couleurs avec les étagères, en fonction de combien la library a d'étagères.
  - Création des loggers pour chaque classe, et ajout des bibliothèques correspondantes dans le pom.
  - Création de la fenêtre de base de l'IHM avec le bouton de génération et les quatre panels.
  - Création d'un datepicker pour le choix de l'année : abandonné par la suite car ne répondant pas au besoin (le datepicker permettait de choisir une date complète, pas seulement l'année). Autre idée : la date est un champ pouvant être modifié seulement par des boutons plus ou moins (abandon car trop long pour l'utilisateur).
  - Débogage massif sur la génération des images, en particulier des fonctions drawBooksAndTitle et drawBooks. Exemple : les livres ne s'affichent pas, mais leurs titres si, et à des emplacement inappropriés.
  - Ajout d'une borne inférieure de nbBooksPerShelf dans l'IHM : pas en-dessous de 1.
  - Ajout d'une inclinaison aléatoire des livres : entre 0 et 45° ou complètement aplati en fonction de la place restante sur l'étagère.
  - A l'ajout d'un livre, mise en place de vérifications : les champs first name, last name, et title sont obligatoires. Si les champs année et dimensions ne sont pas renseignés, ou mal renseignés (pas un entier), ajout d'une valeur par défaut.
  - Ecriture du code permettant de remplir la library à partir d'une liste de livres.
  - Ecriture de la javadoc des fonctions créées
  - Enlever toutes les erreurs et warnings avant le rendu final.
- FREMIN DU SARTEL Thibaud
  - Initiation de la classe **[SVGLibrary]** et génération d'une image SVG (rectangle représentant un livre) avec un titre.
  - Création du fond de la bibliothèque (fond coloré + étagères)
  - Initiation de la classe **[Main]**
  - Génération automatique des couleurs et des tailles pour les livres dont ces attributs n'ont pas été rentrés par l'utilisateur.
    - Remarque : Cette fonctionnalité ne sera finalement pas utilisée car cela sera géré à partir des options (choisies sur la fenêtre d'accueil par l'utilisateur).
      Des modifications avaient aussi été apportées dans ce cadre à readFile et writeFile (qui ont ensuite été transformées en la classe dataFile)
  - Dans la classe **[ConnexionToCongressLibrary]**, écriture de la méthode extractData() récupérant le titre, l'auteur et la date de parution dans le fichier MarcXML converti en String.
    - Remarque : La Méthode utilise le String renvoyé par la méthode getMarcXml() codée par Merlène. J'ai donc directement utilisé les méthodes de la classe String, et non les méthodes de parsing que l'on peut trouver sur XML.

- ➢ Ecriture de la Javadoc

- ❖ JAMMES Fanny:
  - ➢ Génération des documents SVG à l'aide de la bibliothèque Batik **[SVGLibrary]**
    - ■ Difficulté rencontrée : Appropriation de la bibliothèque (complexe, nécessite beaucoup de documentation)
  - ➢ Génération des livres en face-view avec implémentation des orientations futures (Avec possibilités d'insérer des informations tel que le titre et la date, une couleur…) **[SVGLibrary]**
  - ➢ Modification du modèle : Aucune modification n'a été retenue suite à la réunion avec M.Cailloux (Taille des livres, taille de la librairie, taille des étagères) [diverses classes package model]
  - ➢ Mise en place de l'IHM de l'application : Frame générale, sous-frame & composants **[Windows2DLibrary]**
  - ➢ Adaptation du code au niveau des ressources pour rendre la génération SVG polymorphe
    - ■ Note personnelle : Une approche plus exhaustive tel que la demande de sauvegarde dans un dossier mentionné par l'utilisateur aurait été un gros "+". J'aurais vraiment aimé développer cette idée mais par manque de temps et dans un soucis de méthode agile (présentation fonctionnelle à chaque séance) je me suis rabattue sur un répertoire par défaut. Enhancement project for next students ?
  - ➢ IHM : Mise en place des Listeners pour la régénération de la librairie **[Windows2DLibrary]**
  - ➢ Convertion d'un fichier SVG en JPG **[SVGLibrary]**
    - ■ Difficulté rencontrée : Nécessite une grande documentation préalable, beaucoup de recherche quant aux dépendances maven. La génération d'un fichier PNG (vide..) a été plus aisément acquise que la récupération de toutes les données du SVG pour en faire un PNG (conversion intégrale)
  - ➢ Update de l'affichage de la librairie (jpg) après chaque génération dans le JPanel **[Windows2DLibrary]**
    - ■ Difficulté rencontrée : Problème de mémoire ? Reload intempestif de l'image de départ malgré sa suppression et la bonne génération d'une nouvelle image dans le dossier spécifié… Problème résolu après de nombreuses heures de galère avec la génération de nouveau nom de fichier + traitements dans le code
  - ➢ Evolution IHM diverses (add-on, JScroll …) **[Windows2DLibrary]**
  - ➢ Javadoc des méthodes créés
  - ➢ Cleaning package view avant rendu

- ❖ EL CHARTOUNI Elie
  - ➢ Création des fonctions de lecture et d'écriture :
    - ■ Dans un premier temps, la fonction de lecture lisait un fichier texte (.txt) et ajoutait les informations du livre à la bibliothèque. La fonction d'écriture permettait d'ajouter toutes les informations du livre dans un fichier texte (.txt).
    - ■ Ensuite ces deux fonctions ont évolué pour lire et écrire dans un fichier délimité (.csv).
  - ➢ Affichage du livre (position droite) dans une bibliothèque avec le nom de l'auteur et le titre écrit sur l'arête. La taille de police est adaptée en fonction de la longueur du titre et la taille du livre.
  - ➢ Création des fonctions de génération automatique des couleurs et des tailles des livres dans le cas où ces informations ne sont pas saisies par l'utilisateur. Ceci à entraîner une modification dans les fonctions de lecture et d'écriture de fichier.
  - ➢ Gestion de la taille de police du titre pour être centrée dans le livre.
  - ➢ Création d'un nouvel onglet dans l'interface pour l'ajout d'un livre et l'utilisation de la fonction « addLine » dans « DataFile »
  - ➢ Création d'un nouvel onglet dans l'interface pour supprimer un livre de la Librairie. Le programme affiche la liste des livres disponibles. L'utilisateur choisit le livre à supprimer de la liste.
  - ➢ Ajout de l'option de recherche d'un livre sur internet dans l'onglet « Add Books to my Library » et remplissage automatique des champs avec les données trouvées.

- ➢ Mise à jour automatique de la Librairie dans les deux onglets « Add books to my library » et « Delete books from my library » lors de l'ajout ou la suppression d'un livre et ajout de l'option scroll dans les onglets.

❖ CHUNG Hugo

- ➢ Création de la class DataFile et des fonctionnalités qu'elle propose (lecture et écriture)
- ➢ Ajout de la fonctionnalité de tri dans la classe Library
- ➢ Création du visuel de la library dans la classe SVGLibrary. Le gros du travail étant les fonctions de prédictions de positions des libres (en fonction de la place restante l'étagère dans laquelle le placer). Ce travail se faisant indépendamment de l'objet library, c'est à dire que je définissait seulement un nombre de livre à représenter souhaité.

Ce travail a ensuite été complété, et je suis de nouveau intervenu dessus afin de régler des bugs faisant suites aux interventions d'autres dessus. C'était une partie assez compliquée, dans le sens où l'ajout de nouvelles fonctionnalité par dessus mon code a été fait sans vraiment prendre en considération des variables, ou de mes fonctionnalités passées.

- ➢ Création des fonctions de hashcode.

❖ LEJEUNE Merlène

- ➢ **[2D_library/pom.xml]**
  Implémentation du projet Maven (structure, pom, imports nécessaires, groupId et artifactID corrects), bon nommage des packages, des classes et des méthodes.
- ➢ **[model]**
  Initiation du modèle, création des classes Author, Book, Shelf et Library.
  - ■ **Difficultés rencontrées :** comment modéliser correctement la bibliothèque, est-ce qu'une bibliothèque se caractérisent par des étagères qui contiennent des livres ou directement par une liste de livre ? On a choisis la première option.
- ➢ **[model/Library.java] sortByYear(boolean rising) , sortByAuthor() , sortByTitle()**
  Implémentation des méthodes de tri de la bibliothèque (tri par date, auteur et titre)
- ➢ **[view/SVGLibrary.java] drawBooksAndTitles(…) , drawBook(…) , drawTitle(…)**

Affichage avancé des livres dans la bibliothèque : dernier livre de chaque étagère qui s'incline si l'option est sélectionnée, correspondance de la forme du livre avec son titre et son auteur.

- ■ **Difficultés rencontrées :** je pense que de créer des formes de livre séparément des objets Book n'était pas optimal vu les modifications qui ont été faites.

drawBackOutlines(…), drawShelves(…) , drawBook(…) drawTitle(…)
Ajout des différentes nuances de couleur selon l'option (Auto/Light/Dark) et considération de la couleur d'un livre si prédéfinie pour l'affichage.

- ➢ **[controller/ConnexionToCongressLibrary.java] getMarcXML.java**
  Création de la connexion REST afin de récupérer les données d'un livre sur Internet en MarcXML (on suppose que l'utilisateur connaît l'identifiant du livre).
- ➢ **[2D_library]**
  Nettoyage du code : suppression des variables globales, des classes / méthodes statiques, des variables / méthodes inutiles, création d'état dans les classes n'en possédant pas.

Ecriture de la javadoc du modèle et du controller.
Corrections de bugs importante.

- ➢ **[view/Windows2DLibrary.java] getPanelCentreOptions() + classes Listener en bas du fichier**
  Implémentation des boutons et des listeners pour les options de la bibliothèque (couleur des livres, des étagères, du fond, si on active le mod "penché", le mode tri, le nombre de livres par étagère).
- ➢ **[2D_library/src/test/java/io/github/oliviercailloux/y2017/my_2D_library/]**
  Implémentation des tests unitaires du modèle et du controller (sauf pour les testEqual codés par Olympie).