Rapport PIIA : MergeMate

Meriem ERNEZ – Jessy FALLAVIER

Introduction :

Notre projet est une application Java s'appelant MergeMate permettant d'étudier les variations d'un texte. En effet, lors de l'exécution de l'application, on a la possibilité de choisir entre soit deux fichiers (référence et modifié) ou un fichier état regroupant les actions effectuées. Après avoir choisi, nous nous retrouvons sur une page affichant les deux fichiers avec leurs modifications visibles (avec un code couleur) et de nombreuses fonctionnalités.

Ce projet a pour but de pouvoir faciliter la modification d'un texte collaborativement en gardant une trace de ces modifications.

Organisation :

Pour ce projet, nous nous sommes réparti le travail et nous avons récupérer le travail de chacun à l'aide d'un GIT commun.

En effet, pour la réparation du travail, chacun avait plus de facilités avec certaines notions que l'autre. Ainsi, nous avons essayé de nous répartir le travail équitablement par rapport aux facilités de chacun.

Pour ce projet nous avons utilisé la structure d'un projet java simple :

- Le répertoire src contient les différentes classes java. Nous avons utilisé le format MVC.
- Le répertoire res contient des fichiers txt et json
- Le répertoire config contient les fichiers pouvant nécessaires à l'application
- Le répertoire lib contient les librairies java utilisées

Fonctionnalités :

Toutes les fonctionnalités du cahier des charges ont pu être réalisées.

Les fonctionnalités de notre application sont :

- Le chargement de fichiers txt de référence et de modification contenant jusqu'à 1000 signes.
- La visualisation côte à côte des deux textes, la distinction des différents types de modification
- La validation ou le refus d'une modification
- Les commentaires des modifications
- L'édition libre des modifications, la distinction des contre-propositions
- La visualisation / sauvegarde du fichier résultat
- L'enregistrement / le chargement d'un fichier représentant l'état de la session de travail
- La sauvegarde rapide (pour fluidifier le processus de sauvegarde)
- Le Mode sombre/clair sur l'application
- L'annulation / rétablissement les actions réalisées

Implémentation effective :

Le chargement de fichiers txt de référence et de modification contenant jusqu'à 10000 signes.

L'utilisateur a la possibilité de choisir chaque fichier depuis une boîte de dialogue de sélection de fichier. Il ne peut choisir que des fichiers .txt grâce à un filtre.

Lors du choix, on vérifie qu'un fichier a bien été sélectionné et qu'il comporte moins de 10 000 caractères.

Si le choix est validé, le nom du fichier est indiqué dans le bouton. Sinon, l'erreur correspondante est indiquée. La couleur est également modifiée en conséquence.

La visualisation côte à côte des deux textes, la distinction des différents types de modification

On fait appel à la librairie Java diffutils pour qu'elle compare deux textes. Cette dernière va nous rendre deux textes avec des ~ lors d'une suppression dans le fichier de référence et des * lors d'un ajout dans le fichier de modification. Grâce à un algorithme, nous pouvons déduire s'il s'agit d'un changement, d'une suppression, d'une insertion ou d'un non-changement. Une liste de tokens est donc générée. Notons que qu'un espace, un point, une virgule ou un de ces deux derniers suivis d'un espace peuvent faire la jonction entre deux tokens de type changement.

La couleur de chaque token suit un code couleur. Il est noir s'il ne s'agit pas d'une modification, orange s'il s'agit d'un changement, rouge s'il s'agit d'une suppression

et vert s'il s'agit d'une insertion. Les tokens de changement sont accompagnés d'un compteur pour mieux se repérer.

Nous avons la possibilité de cliquer sur chaque token. Le token devient alors surligné en jaune. Pour désélectionner un token, il nous est possible de recliquer dessus ou de cliquer sur une partie de texte non modifiée.

Si la modification cliquée n'est pas dans le champ de vision d'une page, on la téléporte à l'endroit de la modification.

La validation ou le refus d'une modification

Deux boutons deviennent cliquables lors d'un clic sur une modification. Ces deux boutons permettent de valider ou de refuser une modification. Il est possible d'annuler ce choix en recliquant sur le bouton.

Les commentaires des modifications

Lors d'un clic sur une modification, un champ de texte devient éditable. Ce champ de texte permet d'ajouter un commentaire sur la modification si nécessaire. La sauvegarde du commentaire est automatique.

Un token commenté est accompagné d'une petite icône supérieure pour le distinguer.

L'édition libre des modifications, la distinction des contre-propositions

Chaque modification est susceptible d'être contre-éditée. Pour ce faire, il suffit de cliquer sur une modification tout en appuyant sur Ctrl.

Ainsi, une fenêtre modale s'ouvre, contenant un champ de texte éditable permettant de modifier le texte, un bouton pour supprimer la contre-édition et un autre pour la confirmer.

Le texte modifié est soit celui de référence, soit celui de la modification, en fonction de l'endroit où l'utilisateur a cliqué. Si l'utilisateur modifie un token "non modifié", la modification s'appliquera des deux côtés. Un surlignage bleu sera ajouté pour indiquer l'endroit qui sera contre-édité.

Cette contre-édition est distinguable des autres types de modifications car le texte contre-édité sera souligné en bleu.

La visualisation / sauvegarde du fichier résultat

Une fois que toutes les modifications ont été traitées, le fichier résultat peut être sauvegardé au format .txt. Un compteur indique le nombre de modifications restantes à traiter. L'utilisateur peut choisir, grâce à une invite de choix de fichier, l'endroit où ce fichier sera sauvegardé.

Pour créer ce fichier, nous parcourons tous les tokens. Si un token est validé, nous conservons la version du fichier modifié. Si un token est refusé ou s'il s'agit d'un "non-changement", nous conservons la version du fichier de référence.

Un aperçu est disponible, montrant le contenu du futur fichier résultant. Les modifications non traitées apparaissent comme à définir.

L'enregistrement d'un fichier représentant l'état de la session de travail

Nous avons la possibilité de sauvegarder l'état de la session au format JSON. L'utilisateur peut choisir l'emplacement de sauvegarde grâce à une invitation de choix de fichier. Nous utilisons la bibliothèque Gson pour créer ce fichier JSON.

Ce fichier d'état contient l'ensemble des tokens de modifications, avec leur type, leur statut validé, leur texte de référence/de modification, ainsi que leur version contre-éditée et leur commentaire. Il contient également la position de ces tokens dans les fichiers de référence et de modification.

Ce fichier est importable lors du lancement de l'application.

La sauvegarde rapide (pour fluidifier le processus de sauvegarde)

Dès que le fichier résultat ou le fichier d'état est sauvegardé, son chemin est enregistré. Ainsi, pour effectuer la même sauvegarde ultérieurement, il suffit de sélectionner cette option.

De plus, cette action peut également être réalisée en appuyant sur Ctrl + S.

Le Mode sombre/clair sur l'application

De nombreux utilisateurs préfèrent avoir un mode sombre sur les applications qu'ils utilisent. C'est pourquoi sous le bouton "Interface" du menu, on retrouve un bouton pour activer le mode sombre. Une fois activé, l'interface adopte un fond noir et les boutons changent de couleur pour être plus foncés. En retournant dans le menu, on remarque que le bouton "Mode sombre" est devenu "Mode clair", permettant d'inverser l'action pour revenir à une interface claire.

L’annulation / rétablissement les actions réalisées

Nous avons la possibilité d'annuler et de rétablir les actions réalisées grâce à un historique. À chaque action, un état est sauvegardé (jusqu'à un maximum de 10 à la fois).

De plus, ces actions peuvent également être effectuées en appuyant sur Ctrl + Z pour annuler et Ctrl + Y pour rétablir.

Conclusion et perspectives

En conclusion, la création de MergeMate a été un projet ambitieux et passionnant pour nous. Notre application Java offre une solution efficace pour étudier les variations d'un texte de manière collaborative tout en conservant une trace claire des modifications apportées.

Cependant, nous avions envisagé l'ajout d'un bouton "reprendre là où on était" pour permettre aux utilisateurs de retrouver facilement leur position précédente dans le processus de modification du texte. Malgré les difficultés que nous avons rencontrées, nous sommes satisfaits avec notre travail et nos fonctionnalités.