

TP Java POO

Application d'entraînement au vocabulaire



Table des matières

<i>Section 1 Présentation de l'application.....</i>	<i>3</i>
1.1 But.....	3
1.2 Énoncé.....	3
<i>Section 2 Regarder la maquette fournie.....</i>	<i>5</i>
2.1 But.....	5
2.2 Énoncé.....	5
<i>Section 3 Travail à effectuer.....</i>	<i>6</i>
3.1 But.....	6
3.2 Énoncé.....	6
3.2.1 Conception.....	6
3.2.2 Réalisation.....	7
3.2.3 Livrables.....	7
3.3 Utilisation d'une BD – Utilisation de Hibernate.....	7

Section 1 Présentation de l'application

1.1 But

- Prise de connaissance du cahier des charges

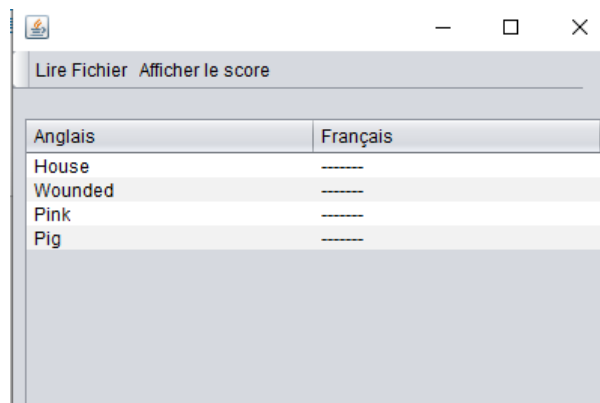
1.2 Énoncé

L'application propose, sous forme d'un jeu, de mettre en correspondance des mots d'une langue 2 par rapport à des mots d'une langue 1.

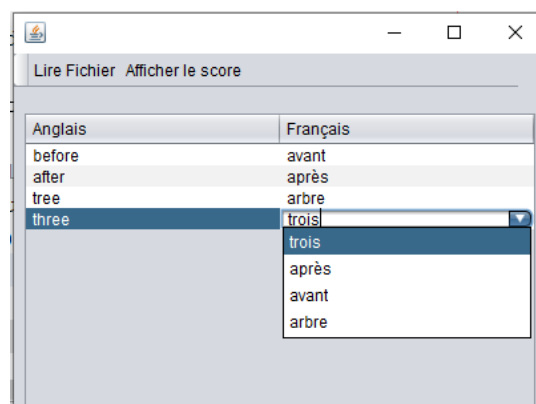
Il peut donc être utilisé, par exemple, pour retrouver les mots français à partir de mots anglais. Mais la même application peut proposer en langue 1 de l'allemand, en langue 2 de l'italien ...

Une maquette en java est fournie. Voici sa présentation :

Le jeu à son lancement :



En cours d'utilisation, la colonne de droite de la grille dispose d'une liste à choix multiples (ComboBox) :



L'application se compose d'une fenêtre principale.

En partie haute, un bouton « Lire fichier » permet de rechercher un fichier contenant les mots langue 1 et mots langue 2.

Le contenu de ce fichier permet alors de définir le titre des deux colonnes, et les mots qu'il faut mettre en relation.

Dit autrement, le fichier contient le nom des 2 langues, ces noms seront affichés en tête de colonne ; les autres informations du fichiers constituent des couples de String, une valeur pour la colonne de gauche, une valeur correspondante pour la colonne de droite.

La présentation des mots à trouver dans la colonne de gauche est dans un ordre aléatoire.

Le bouton « Afficher le score » ouvre une fenêtre de score. Elle indique le nombre de correspondances trouvées.

Le bouton « rejouer » (non représenté ci-dessus) permet de redistribuer une partie des mots à trouver, en privilégiant ceux qui ont été non trouvés les fois précédentes. Pour autant des mots déjà trouvés peuvent faire partie de cette liste.

Section 2 Regarder la maquette fournie

2.1 But

- Comprendre du code existant

2.2 Enoncé

Une maquette en Java est fournie.

Elle a été faite avec l'éditeur NetBeans et avec la couche graphique Swing.

Les couples langue1 , langue 2 sont dans cette maquette en dur dans le code, pour essai.

Vous allez repenser cette application en JavaFX.

Section 3 Travail à effectuer

3.1 But

- Concevoir en objet

3.2 Enoncé

3.2.1 Conception

Nous allons essayer d'orienter la conception pour avoir 2 packages côté code :

- un package « model », qui gère les données de l'application et les méthodes « métier » utilisable dans l'autre package
- un package qui gère le code main de l'application et les contrôleurs

3.2.1.1 La couche Model

Réfléchir à une classe qui offre une méthode pour lire un fichier, et analyser son contenu.

Le fichier peut être en xml, du json ou plus simplement sous la forme csv

exemple avec le format csv:

```
anglais ; français
tree ; arbre
for ages ; depuis des lustres
xxx ; yyy
```

Dans cette même classe proposer une méthode :

- pour récupérer le titre des colonnes
- pour récupérer les mots langues 1 ... ou liste d'objets
- pour récupérer les mots langues 2 ... ou liste d'objets

Il peut être utile d'imaginer une autre classe qui contiendrait les mots langue 1 / langue 2, le « taux de succès » lors d'un précédent jeu ...

La sérialisation d'objets peut être aussi une autre piste ...

Faire un diagramme de classe.

3.2.1.2 La couche View

Elle sera réalisée directement en utilisant Scene Builder.

3.2.2 Lot 1 : Réalisation

Coder l'application en Java FX.

3.2.3 Livrables

Le projet Java est à fournir dans un zip

Une copie d'écran du diagramme de classe est à placer dans les ressources du projet.

Si souhaité vous pouvez joindre un document pdf contenant votre réflexion.

Le zip porte votre nom et est à envoyer à

contact@pragma-tec.fr

3.3 Lot 2 : Utilisation d'une BD

Faire un commit du code du lot 1.

L'application évolue :

Dès que l'on quitte l'application, les « taux de succès » de la personne sont perdus.

On pourrait les mémoriser dans le fichier txt, mais on décide de le faire par une BD.

Cette BD doit être conçue en multi utilisateurs :

Le jeu de données n'est pas identique à tout le monde, un joueur peut s'entraîner pour de l'allemand/français, un autre pour de l'anglais/français ...

Les scores sont propres à chaque utilisateur. L'outil proposera donc les mots à trouver en fonction des précédentes réponses.

Revoir la couche Model en utilisant une BD MySQL

Concevoir un MCD avec Jmerise ou équivalent de la Base.

Un dialogue permet de se connecter à la BD. Idéalement, lors du 1^{er} emploi, la création de la BD et des tables devrait se faire par cette application (pas de fichier SQL séparés).

Un dialogue dans l'IHM permet d'écrire le contenu du fichier csv en BD.

L'application utilise ensuite les données BD, avec lecture du taux de succès de chaque couple langue1, langue2 ...