



Rapport de développement/test.

Par MOUNIER Dorian

Sommaire

I - Synthèse du développement : Page 3-4

II - Description des test unitaires réalisés : Page 4-5

Synthèse du développement

L'objectif initial de notre projet était de faire un glossaire fonctionnel avec deux vues différentes avec la possibilité aux étudiants de pouvoir se tester à l'aide d'un autotest qui se génère et se corrige seul sur les pages souhaitées (avec ou sans les mots demandés au DS), la possibilité que les étudiants voient le glossaire complet ou les pages souhaitées puis finalement que les étudiants puissent proposer des mots au professeur.

La vue professeur devait posséder plusieurs fonctionnalités :

- la possibilité de voir et de faire les tests comme les étudiants
- de voir les mots proposés par les étudiants pour pouvoir les ajouter au glossaire
- de générer des tests
- de modifier le glossaire (enlever et rajouter des mots)

Les fonctionnalités que nous n'avons pas implémentées sont :

- "éviter un mot trop courant, ou trop dur, ou équilibrer la difficulté des deux sujets" [cahier des charges] Cela n'a pas été implémenté suite à une conversation avec madame Metge qui nous a expliqué qu'elle ne voulait pas implémenter des difficultés différentes en fonction des mots.
- le droit admin car dans notre idée de départ ce statut permettait de faire une gestion globale de l'application mais finalement cette idée n'a pas été retenue car la vue professeur suffit à gérer le glossaire et les problèmes techniques sont résolus dans le code source.

Pour utiliser l'application, il suffit de lancer la classe Main qui ouvrira une première fenêtre avec un bouton et une zone de texte.

La zone de texte permet de rentrer un pseudo, nom, prénom,... (il faut penser à effacer le texte par défaut de l'application qui ne se supprime pas seul). (

Ensuite, deux choix s'offrent à nous: Soit notre login (pseudo expliqué ci-dessous) est metgec et un mot de passe est demandé.

Si notre login est différent de metgec, nous entrons dans la vue étudiant.

La vue étudiant donne accès au glossaire avec la possibilité de lire les pages du glossaire de notre choix: (From page ... to ...) pour choisir la première page jusqu'à la dernière (les deux pages de délimitation comprises). Il faut ensuite cliquer sur le bouton "Run" pour lancer la visualisation des pages saisies. Il est possible de changer les pages visualisées en changeant les chiffres dans la même zone que précédemment et en cliquant sur le bouton "Run".

La vue étudiant donne également accès à l'onglet "Quiz", où l'on choisit les pages sur lesquelles nous voulons nous entraîner comme précédemment. Il y a aussi une option active par défaut pour être testé uniquement sur les mots sur lesquels nous sommes évalués en DS (il est possible de désactiver cette option en cochant la case correspondante). Après avoir sélectionné les pages sur lesquelles nous aurons un test, il faut cliquer sur le bouton "Play". Cela actualise la page et nous permet de commencer le test avec 20 mots sélectionnés aléatoirement présents dans les pages sélectionnées par l'utilisateur. Il faut simplement écrire la traduction en anglais du mot dans la case adjacente au mot français. Lorsque l'on considère que l'on a terminé le test, nous cliquons sur le bouton "Confirm" (ce bouton est disponible

même lorsque toutes les cases ne sont pas remplies), qui corrigera avec la traduction attendue et donnera une note sur 20 en fonction de nos réponses (avec 1 point par réponse juste). Il est possible de relancer le test avec ou sans changer les pages ou les options en re cliquant sur le bouton “Play”.

Dernier élément de la vue étudiant, nous pouvons accéder à l’onglet “Proposal” qui permet de proposer un ajout de mot en écrivant le mot en anglais et sa traduction avec les deux zones de texte et en cliquant sur le bouton “Create a proposal”. Cette proposition sera envoyée à la vue professeur (voir vue professeur). Une fois que l’on a cliqué sur le bouton “Create a proposal”, une petite fenêtre nous montre que notre proposition a bien été sauvegardée (“Proposal Saved”). A partir d’ici, il suffit de cliquer sur le bouton “OK” et nous pouvons continuer d’utiliser l’application.

Lorsque l’on quitte l’application (croix en haut à droite ou autre méthode) une petite fenêtre s’ouvre pour nous demander si l’on veut sauvegarder le glossaire (utile uniquement dans la vue professeur car la seule vue où l’on peut modifier ce dernier).

Passons maintenant à la vue professeur. Comme dit précédemment si le login est metgec un mot de passe est demandé. Après avoir inscrit notre mot de passe (“admin” pour le test) dans la zone de texte, il faut cliquer sur le bouton “Connect as teacher”. Si le mot de passe n’est pas rentré ou n’est pas valide, l’application ne s’ouvre pas.

Une fois dans la vue professeur, on peut comme pour la vue étudiant voir le glossaire et faire des quiz.

L’onglet “Proposal” permet de voir les mots proposés par les étudiants et de les ajouter en cliquant sur le bouton “Add a Word”.

L’onglet “Test Generator” permet de générer des tests en format HTML via le bouton “Export” avec le choix des pages, l’option des mots présents en DS, le nombre de mots, le nom du sujet (sujet numéro 1, permet d’avoir le titre pré-saisi dans le fichier) et la valeur des réponses (+/- n points).

Finalement, l’onglet “Modify Glossary” permet de modifier le glossaire :

- Nous pouvons rechercher des mots dans le glossaire en l’écrivant dans la première zone de texte (sur la première ligne) pour pouvoir supprimer ce mot à l’aide du bouton “Delete this word”. une fenêtre va alors s’afficher et il suffit d’appuyer sur le bouton “OK” pour que la page disparaisse.

- On peut aussi ajouter un mot en écrivant le mot en anglais dans la première case de la seconde ligne et sa traduction française dans la case en face puis en cliquant sur le bouton “Add to Glossary”. Le mot sera ajouté au glossaire (on peut le rechercher pour tester via la fonctionnalité présentée ci-dessus).

Description des tests unitaires réalisés

Des tests ont été réalisés sur toutes les fonctionnalités du glossaire pour vérifier si elles fonctionnaient comme prévu. Si aucune remarque n'est précisée à la fin du déroulement d'un test, alors l'issue du test a été positive.

Nous avons testé les mots de passe via la classe `PassWordTest` en essayant différentes chaînes de caractères.

Pour cela, nous avons fait passer les chaînes de caractères test dans la méthode `toHexa` puis nous récupérons la nouvelle chaîne de caractères obtenue dans la méthode `toMD5Hash` pour enfin comparer la chaîne de caractères obtenue avec le **résultat attendu** (la chaîne de caractère "admin" en MD5). Nous faisons ensuite trois tests: Un avec le bon mot de passe ("admin") à l'aide de `assertEqual`, un deuxième avec "Admin" pour tester si la détection des majuscules fonctionne avec un `assertNotSame` (permet de vérifier que le résultat obtenu est différent du résultat du vrai mot de passe), et un troisième avec une chaîne de caractères aléatoire pour finir le test avec un `assertNotSame`.

Nous avons fait un nouveau test afin de vérifier les méthodes de la classe `Glossary` (sauf `sortByEn` car elle est utilisée dans `addWordGlossary` testée après et `setGlossary` car elle est très simple [une seule ligne `this.glossary = glossary`], `pages` et `pagesMandatory` car ces deux dernières méthodes renvoient des List de mots et cela nous a semblé trop long et trop complexe).

Nous avons commencé par initialiser un mot `RA` ("Hello", "Bonjour") qui sera utilisé plusieurs fois durant ce test et un `BadResult` ("1", "1") pour tester avec un autre résultat incorrect. Ensuite, nous initialisons un Glossaire de test nommé `Test1` puis on recherche si le mot ("Hello") est présent à l'aide d'un `equals` et d'un `assertEquals` entre la valeur du `equals` et de `false`. Ensuite, nous utilisons la méthode `addWordStringOnly` et on ajoute au glossaire `Test1` deux chaînes de caractères ("Hello", "Bonjour") puis on recherche ("Hello") dans le glossaire (avec la méthode `search`) et on le compare au mot `RA` (le résultat attendu).

Avec la méthode `assertEquals`, on teste le nombre de pages avec la méthode `numberOfPages` et avec une comparaison à l'entier 1, puis on teste des résultats faux pour voir si notre programme ne répond pas toujours vrai (à l'aide de la méthode `assertNotSame`). On teste donc cela avec la recherche du mot "Hello" avec `BadResult` initialisé au début et un nombre de page égal à 5. On retire ensuite le mot `RA` du glossaire, on vérifie qu'il n'y est pas (que la suppression a bien fonctionné [même test que lorsque le mot n'avait pas été ajouté]).

Nous avons ajouté le mot `RA` avec une autre méthode `addWordGlossary` qui ajoute des mots et non deux chaînes de caractère puis nous avons fait les mêmes test pour voir si le mot était présent dans le glossaire et pour voir s'il ne renvoyait pas toujours vrai.

Pour terminer, on enlève une dernière fois le mot `RA` et on teste pour voir s'il n'y est effectivement pas.