



École des Ponts

ParisTech

École des Ponts ParisTech

Pour le 25 janvier 2016

Anna Le Cerf
Marion Tormento

Techniques de développement logiciel ANMA, Logiciel d'aide à la révision

Introduction

Dans le cadre de la réalisation du projet de Techniques de Développement Logiciel, nous avons décidé de réaliser un logiciel d'aide à la révision. Le principe est simple : l'utilisateur prépare des fiches d'interrogation, de vocabulaire par exemple, puis peut se faire interroger sur les thèmes qu'il a conçus. Ce logiciel s'adresse principalement aux préparateurs qui souhaitent réviser efficacement leurs examens.

Objectifs initiaux

Nous avons pour objectif initial de créer un logiciel qui puisse interroger l'utilisateur de manière très variée : allant de la simple traduction d'un mot à la proposition de définitions ou de phrases plus complexes, en passant par les formules mathématiques et autres outils utiles aux préparateurs.

La validation de la question de vocabulaire aurait été permise par une comparaison simple entre une proposition aléatoire du logiciel à l'intérieur d'un thème choisi par l'utilisateur et sa réponse : si la réponse de l'utilisateur est identique à la réponse attendue, elle est juste. Les formules mathématiques proposées par l'utilisateur auraient nécessité l'importation d'un module permettant d'écrire des symboles, puis une comparaison semblable à celle du vocabulaire aurait été effectuée. Plus élaborée, la vérification des réponses dans le cas de définition par exemple, aurait impliqué une analyse par mots clés: la réponse de l'utilisateur aurait été validée si l'on y retrouvait une partie des mots-clés prédéfinis.

D'un point de vue structurel, nous voulions proposer deux modes, un mode préparation et un mode évaluation. Dans le mode préparation, l'utilisateur rentre à la main les informations sur lesquels il souhaite être interrogé par la suite en indiquant les questions ainsi que leurs réponses. A première vue rébarbative, cette étape est néanmoins un moyen efficace pour le préparateur de réviser ses leçons. Passé en mode évaluation, l'utilisateur est interrogé sur le thème de son choix. Le logiciel évalue la justesse de ses réponses selon les différentes manières évoquées plus haut puis délivre une note. Une possibilité d'extraire sous PDF les fiches élaborées par l'utilisateur constituait également un de nos objectifs.

Choix Techniques

Nous avons choisi de réaliser un logiciel écrit en langage python. En effet, n'étant que deux dans notre équipe, il était risqué de se lancer dans l'apprentissage d'un nouveau langage à la vue du peu de temps que nous avons pour mener notre projet à bien. De plus, bien que l'idée de concevoir un site internet ait été évoquée un temps, nous n'avons pas choisi cette voie car le fait d'utiliser une connexion nous apparaissait plus comme un défaut qu'un atout. Nous avons donc besoin de réaliser une interface graphique et avons choisi d'utiliser le module PyQt4 que nous avons déjà manipulé lors d'un TP. Afin de partager le plus efficacement possible nos données, nous avons toutes deux installé Git, et avons échangé nos codes grâce à GitHub en les déposant sur le repository accessible à cette adresse <http://www.github.com/MarionTormento/ProjetTDL0G>.

L'architecture Model-View-Controller s'est imposée comme une évidence. Le code a donc été séparé en deux fichiers: un code Model et un code View; le code Model est importé dans le code View.

Le code Model Dans ce code a été importé le module random, pour le choix aléatoire des questions lors de l'évaluation, ainsi que les modules PyQtGraph et numpy afin de réaliser des graphiques pour les statistiques de l'utilisateur. Ce fichier comprend le code de base pour l'écriture, la lecture et l'interrogation des fiches, la comparaison des réponses utilisateur aux réponses attendues et la gestion de l'ensemble des fiches et des scores associés.

Le code View Dans ce code a été importé le module QtGui, QtCore et QtCore.Signal de PyQt, les modules pyqtgraph et numpy pour les réalisations de graphes, le module os ainsi que le code Model. Le code View comprend l'interface graphique qui propose un accès aux fenêtres Préparation, Evaluation et Statistiques. Des systèmes de QPushButton et de QComboBox connectés permettent l'ouverture et la fermeture des différentes fenêtres.

Réalisations & Méthodologie

Organisation

Afin d'assurer le bon déroulement de notre projet, nous avons décidé de séparer code View et code Model. Anna s'est donc penchée sur l'élaboration de l'interface graphique du logiciel tandis que Marion s'est concentrée sur le code du fonctionnement à proprement parler du logiciel proposé. Loin de nous cantonner à nos tâches respectives sans partager nos travaux, cette organisation partitionnée a nécessité une grande communication entre nous afin que l'importation finale du code Model dans le code View puisse se dérouler sans trop d'encombres.

Etapes de réalisations

Interrogation de vocabulaire

Nous avons donc débuté notre travail en nous penchant sur le code de la préparation et de l'interrogation de vocabulaire d'une langue vers une autre. En effet, aux vues de nos objectifs initiaux, l'interrogation de vocabulaire apparaissait comme la première étape charnière qui nous permettrait de construire l'architecture du code à partir d'un principe "simple" avant de poursuivre sur des situations plus compliquées comme la comparaison de formule mathématiques, puis la comparaison de phrases.

Objectifs révisés

Au fur et à mesure de notre avancement, nous nous sommes confrontées à plusieurs problèmes alors que notre objectif initial nous paraissait simple de prime abord. Tout d'abord, un mot peut détenir plusieurs sens donc plusieurs traductions dans une langue étrangère. Nous avons contourné cette subtilité du langage en supposant que le préparatoire dispose de thèmes bien précis d'étude. Par conséquent, à un mot ne peut correspondre qu'un seul autre, en accord avec le sujet de l'interrogation. De plus, il nous est apparu que le code du mode d'interrogation n'était pas aussi simple que nous l'avions imaginé. En effet, nous avons pensé dans un premier temps comparer simplement la réponse de l'utilisateur à la correction.

Cependant, peut-on considérer une réponse entièrement fausse si elle ne comporte qu'une faute de typographie, ou même qu'une seule erreur d'orthographe? Nous nous sommes alors orientées vers une méthode de comparaison fondée sur la distance de Levenshtein afin de nuancer nos critères d'évaluation. Ainsi, suivant la valeur de la distance de Levenshtein, on évalue la justesse de la réponse et on modifie le score en conséquence.

Face à ces aléas, et au peu de temps restant, nous nous sommes rendu compte que nous avions visé trop haut : l'implémentation du vocabulaire soulevait de nombreuses questions techniques, nous n'aurions jamais le temps de nous intéresser à des cas plus complexes comme des phrases entières ou des formules de mathématiques.

Améliorations graphiques et implémentation de statistiques

Nous avons donc décidé de nous concentrer pleinement sur la préparation de fiches de vocabulaire et sur leur évaluation afin de produire un logiciel aussi performant que possible sur cette fonctionnalité en apparence restreinte.

D'une part, nous avons tenté de rendre notre logiciel le plus fonctionnel possible. Cet objectif a pu être atteint en utilisant par exemple des listes déroulantes pour choisir la fiche d'interrogation ou encore les langues de préparation ou d'évaluation. La création de statistiques sur les scores des différentes sessions d'évaluation passées par le préparateur est également une des fonctionnalités qui rend notre logiciel pédagogique et ludique.

D'autre part, nous avons souhaité que le logiciel soit attrayant pour l'utilisateur. Nous avons donc soigné l'interface graphique afin d'offrir un logiciel visuellement agréable et intuitif pour l'utilisateur.

Résultats obtenus et perspectives d'amélioration

Résultats obtenus

Nous avons donc réalisé un logiciel d'aide à la révision uniquement axé sur l'évaluation de fiches de vocabulaire dans des langues étrangères.

En mode préparation, l'utilisateur nomme la fiche de vocabulaire qu'il va élaborer et choisit les langues de cette fiche dans une liste déroulante. Nous avons géré le fait que l'utilisateur ne peut pas choisir deux fois la même langue lorsqu'il écrit sa fiche, car cette situation n'aurait pas d'intérêt. Il peut rentrer alors ses mots successivement en appuyant sur le bouton "mot suivant". Lorsqu'il considère que sa préparation est finie, il clique sur "Préparation terminée". Le logiciel a alors enregistré sa fiche avec le nom qu'il a choisi dans un dossier "Fiches" créé lors de la première utilisation du logiciel. Cette fiche répond à un code d'écriture simple : la première ligne comporte les deux langues d'écriture inscrite sous la forme d'un couple "langue1:langue2", puis fait apparaître sur chaque ligne suivante un couple "mot1:mot2". Par ailleurs, une fiche "Récapitulatif", créée lors de la première utilisation du logiciel, recense les informations propre à cette nouvelle fiche sous la forme "nom de la fiche:langue1:langue2".

En mode évaluation, l'utilisateur choisit la fiche sur laquelle il désire être interrogé, ainsi que le sens de traduction, à l'aide de menus déroulants. Les thèmes des fiches proposées ainsi que les langues correspondantes sont issus de la fiche "Récapitulatif". De la même façon que lors de la préparation, il est impossible de choisir deux fois la même langue et seule les langues dans laquelle la fiche est écrite sont proposées. Le logiciel va chercher la fiche souhaitée et élabore une série de questions de vocabulaire aléatoires. L'utilisateur est soumis à ces questions et donne ses réponses. Si celles-ci ne sont pas entièrement correctes, la question, la réponse attendue et la réponse de l'utilisateur sont alors mise en mémoire par le logiciel. A la fin de l'évaluation, le logiciel annonce le score obtenu par l'utilisateur et lui fournit la liste de ses erreurs accompagnées de leur correction. Le score est ensuite inscrit dans la fiche "Récapitulatif" à la suite de la ligne des caractéristiques de la fiche.

Lorsqu'il clique sur le bouton "Mes statistiques", l'utilisateur accède à un graphe de ses résultats sur l'ensemble des sessions d'évaluation qu'il a passées. Il peut ainsi constater

son évolution et ses progrès. Ses résultats sont issus des scores inscrit dans la fiche vocabulaire.

L'ensemble des fonctions réalisées ainsi que le descriptif du fonctionnement des différents onglets sont présentés sur les figures 1 et 2

Perspectives d'amélioration

D'un point de vue général, une des premières améliorations que nous pourrions réaliser serait d'aller jusqu'au bout de nos objectifs initiaux et de proposer les mêmes fonctionnalités pour des phrases et des formules mathématiques. De plus nous n'avons pas tenté de réaliser des exportations au format pdf des fiches. Un autre angle attaque intéressant serait de transposer ce logiciel au langage Android pour proposer une application sur téléphone qui serait plus pratique pour les utilisateurs.

Au regard du logiciel réalisé, de nombreuses améliorations restent possible. Pour l'instant le logiciel ne propose que la gestion des langues de l'alphabet latin - pour l'instant seul l'anglais, le français et l'espagnol sont proposés mais il serait simple de rajouter n'importe quelle langue de l'alphabet latin. Il pourrait être intéressant de se pencher sur d'autres alphabets afin de varier la gamme des utilisateurs. Par ailleurs, il faudrait pouvoir gérer la modification et suppression des fiches. En effet si aujourd'hui l'utilisateur fait une faute de frappe lors de la préparation de sa fiche, il ne peut pas revenir dessus une fois que le couple de mot est enregistré. De plus l'ajout de mot à une fiche déjà existante n'est pas optimisé. Il est possible de rajouter des mots sur une fiche existante mais il faut pour cela passer par l'onglet "Préparation", et réécrire exactement le nom de la fiche que l'on souhaite compléter. En plus d'être compliqué, ce procédé n'est pas entièrement correct car il entraine la réécriture du couple "langue1:langue2" à la ligne à laquelle on commence à compléter la fiche. Enfin, un effort général sur la gestion des erreurs est à réaliser: il faudrait empêcher la possibilité de créer une fiche avec un nom déjà utilisé, de créer une fiche sans avoir choisi les langues, etc.

Figure 1: Description de l'ensemble des fonctions existantes

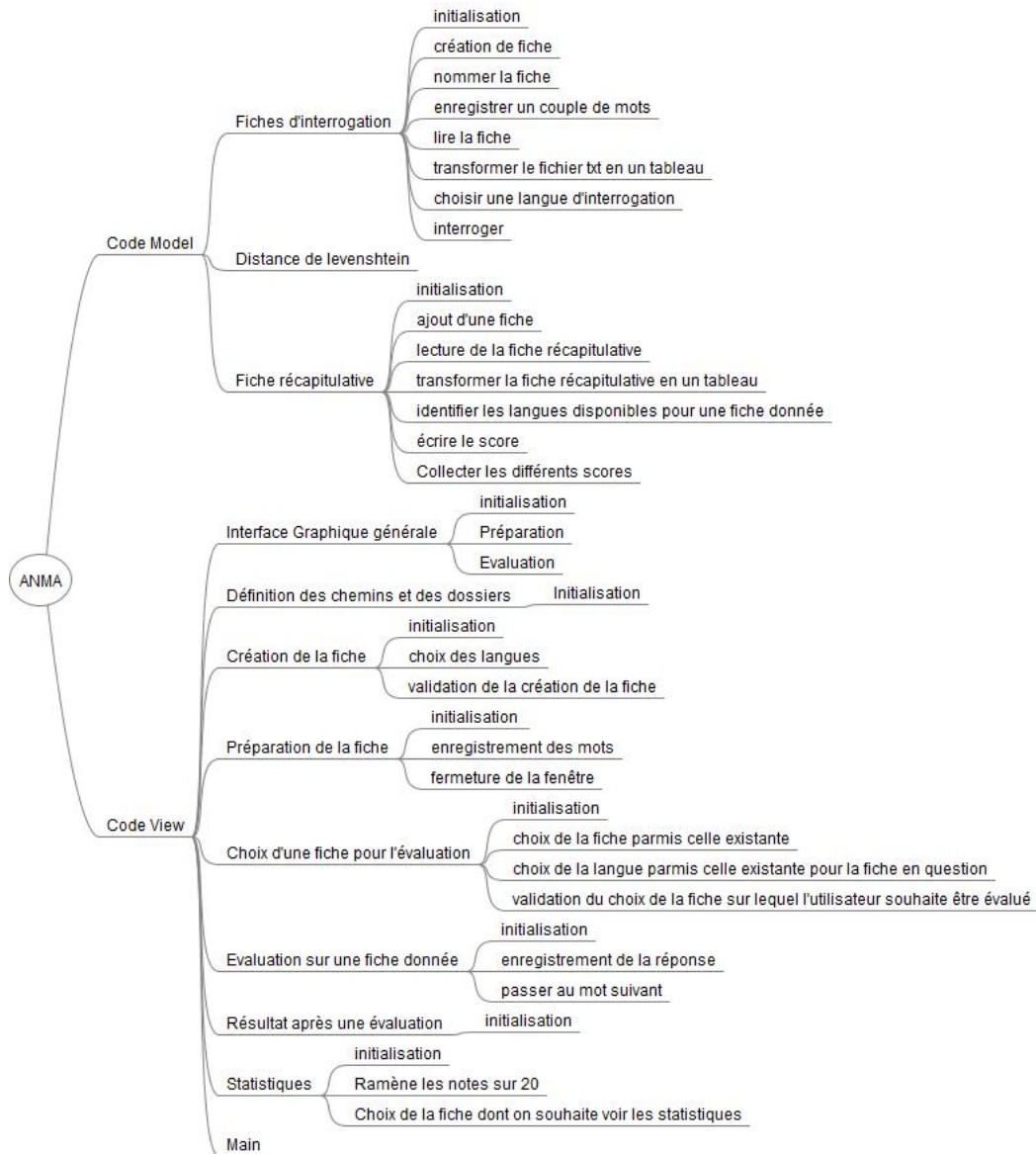
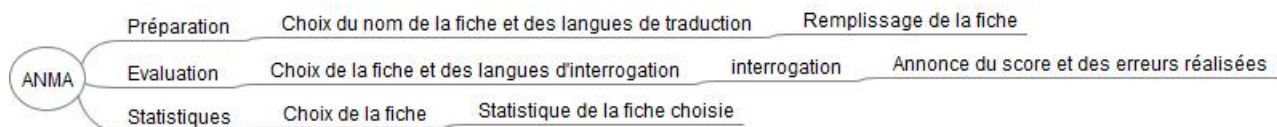


Figure 2: Descriptif du fonctionnement des onglets



Problèmes rencontrés

Parmi les problèmes d'ordre général que nous avons rencontrés lors de la réalisation de notre projet, l'utilisation de GitHub est sans doute celui qui vient à l'esprit immédiatement. En effet, nous avons rapidement compris l'intérêt de cette plateforme de partage, cependant il aurait été bénéfique d'avoir une formation ou du moins quelques notions afin, d'une part, de débiter rapidement son utilisation et, d'autre part, d'éviter des erreurs de partage qui ont pu survenir par la suite.

L'autre problème principal reste la durée impartie pour mener ce projet à bien. Bien que nos objectifs initiaux étaient quelque peu démesurés, il est toujours frustrant de rendre un projet où l'on sait quelles améliorations apporter, comment les réaliser mais manquer de temps pour y arriver.

Tout autre problème mineur que nous avons pu rencontrer a pu être rapidement réglé en ayant recours à des forums comme Stackoverflow ou OpenClassroom.