

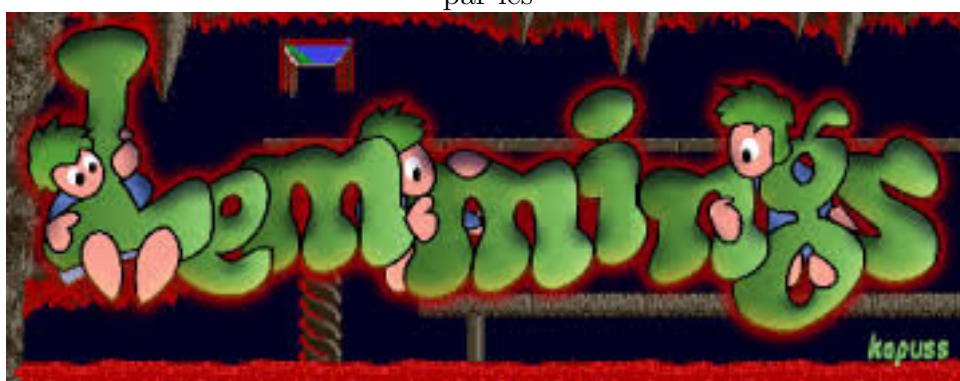
Master AIGLE, IMAGINA et MIT

Générateur semi-automatique de quizz basés sur Wikipedia

avec mise en oeuvre de patterns syntaxiques et construction d'ontologies

TER

Réalisé le 26 avril 2017,
par les



Meryll Essig	meryll.essig@etu.umontpellier.fr
Tristan Cossin	tristan.cossin@etu.umontpellier.fr
Tamara Rocacher	tamara.rocacher@etu.umontpellier.fr
Dorine Tabary	dorine.tabary@etu.umontpellier.fr

Sous la responsabilité de Pierre Pompidor



Université Montpellier
FACULTE DES SCIENCES, SECTION INFORMATIQUE
2 Place Eugène Bataillon, 34090 Montpellier

Table des matières

1	- Introduction	2
1.1	- Le sujet	2
1.1.1	- En phase d'enrichissement	2
1.1.2	- En phase d'exploitation :	2
1.1.3	- Langages/technologies :	2
1.2	- Travail réalisé	3
1.3	- Difficultés rencontrées	3
1.3.1	- Travailler en synergie	3
1.3.2	- Gestion de l'immense quantité d'information de Wikipédia . . .	3
1.3.3	- Apprentissage d'un nouvel environnement MEAN	3
2	- Phase d'études	4
2.1	- Les articles scientifiques	4
2.2	- Gestion de l'extracteur	4
2.3	- Nos choix	4
3	Le programme	5
3.1	Plan général	5
3.2	Côté serveur	5
3.2.1	Extraction des données	5
3.2.2	Génie Logiciel	5
3.3	Côté client	5
3.3.1	IHM	5
3.3.2	Interprétation	5
4	La théorie	6
4.1	intérêt	6
4.2	reformulation en terme de cliques	6
4.2.1	Kesako	6
4.2.2	résolution exacte grâce au branch and bound	6
4.3	reformulation en terme de couverture d'ensembles	6
4.3.1	Kesako	6
4.3.2	résolution exacte grâce aux ϕ -approximation	6
5	conclusion	7
5.1	travail réalisé	7
5.2	Notre approche	7
5.3	ce que le projet nous a apporté	7
5.4	axe de recherche	7

Chapitre 1- Introduction

1.1- Le sujet

Le but du projet est de construire parallèlement des quizz via Wikipédia et une ontologie structurant la thématique des questions posées dans ces quizz.

1.1.1- En phase d'enrichissement

Nous devons :

- récupérer aléatoirement un lot d'articles sous Wikipédia :
 - ne conserver l'article que si :
 - l'article possède au moins une photo d'illustration ;
 - l'article présente une taille (en caractère) minimale à déterminer.
 - extraire via des patrons syntaxiques :
 - la réponse ;
 - la définition directe telle que des définitions dérivées soient aussi construites.
- présenter l'image, la réponse et la définition à un administrateur qui va :
 - vérifier/corriger les éléments présentés ;
 - il faut automatiser la recherche d'autres images si celle présentée est rejetée ;
 - permettre à l'administrateur de valider les définitions dérivées ;
 - attacher l'article à une ontologie (et si nécessaire compléter cette ontologie) ;
 - affecter une difficulté à la question.

1.1.2- En phase d'exploitation :

Nous devons :

- demander au joueur :
 - le nombre de questions (fixé par défaut à 10) et le nombre de thèmes (fixé par défaut à 5) ;
 - faire choisir les thèmes via l'ontologie ou les tirer aléatoirement dans les super-thèmes.
- construire et faire jouer le quizz en présentant les images comme questions :
 - calculer les points à accorder suivant une formule de valuation ;
 - donner la correction après chaque validation ;
- mémoriser le score du joueur ;
- gérer un tableau de bord (statistiques, évolutions de l'expertise du joueur par thème...).

1.1.3- Langages/technologies :

Architecture Mean : MongoDB + Express + Angular + Nodejs.

1.2- Travail réalisé

1.3- Difficultés rencontrées

1.3.1- Travailler en synergie

Notre groupe est composé d'étudiants ayant choisi des formations très différentes (master MIT, AIGLE et IMAGINA).

Plusieurs compétences ont été primordiales comme :

- l'organisation qui a permis le travail de groupe malgré plusieurs emplois du temps très différents ;
- la communication nécessaire pour partager des idées avec d'autres personnes aux formations hétérogènes ;
- l'adaptation inévitable pour transposer nos connaissances au projet.

1.3.2- Gestion de l'immense quantité d'information de Wikipédia

1.3.3- Apprentissage d'un nouvel environnement MEAN

Chapitre 2- Phase d'études

2.1- Les articles scientifiques

2.2- Gestion de l'extracteur

Pourquoi pas MultiWiki/Webedia.

2.3- Nos choix

Chapitre 3 Le programme

3.1 Plan général

3.2 Côté serveur

3.2.1 Extraction des données

Connaissance théorique

Lemmatisation et racinalisation (stemming))

Expressions régulières

Commandes couvrantes

en pratique

3.2.2 Génie Logiciel

Application Web

3.3 Côté client

3.3.1 IHM

3.3.2 Interprétation

Chapitre 4 La théorie

4.1 intérêt

4.2 reformulation en terme de cliques

4.2.1 Kesako

4.2.2 résolution exacte grâce au branch and bound

4.3 reformulation en terme de couverture d'ensembles

4.3.1 Kesako

4.3.2 résolution exacte grâce aux ϕ -approximation

Chapitre 5 conclusion

5.1 travail réalisé

5.2 Notre approche

Nous avons voulu rentabilisé les compétences de chacun et fournir ainsi des perspectives multiples de ce problème.

5.3 ce que le projet nous a apporté

5.4 axe de recherche

Bibliographie

Table des figures