BROSSARD Florian / MILLOTTE Fanny

Projet tuteuré de 3ème année en licence d'informatique

Logiciel d'aide à la réalisation d'algorithmes et génération de code LaTeX

LOGO

Tuteur de projet : M. BOUQUET Fabrice

UFR ST

16 route de Gray
25000 Besançon

Remerciements

Nous remercions Monsieur BOUQUET pour ses précisions sur le sujet, ses conseils, ses idées d'améliorations et pour sa disponibilité depuis le début du projet. Nous le remercions également pour la rapidité et la clarté avec lesquelles il à su répondre aux questions que nous nous sommes poser tout au long de ce projet.

Table des matières

1 - Présentation

1.1 - Contexte

Ce projet a été réalisé durant notre troisième année en licence d'informatique à l'UFR ST de Besançon. Il fait partie des épreuves comptabilisées pour le semestre six de notre cursus et a été créé en binôme.

1.2 - Présentation du sujet

Le but de ce projet est de réaliser un logiciel aidant à la réalisation d'un diaporama représentant un algorithme pour cela le logiciel doit générer du code LaTeX. Le choix du langage de programmation étant libre.

Nous avions une représentation du rendu visuel d'une diapositive (voir si dessous). Nous pouvons ainsi observer le pseudo code en haut à gauche, le tableau des variables en haut à droite, de même que le graphique représentant l'avancement de l'algorithme.

Le code LaTeX généré par le logiciel doit être basé sur beamer pour la gestion des animations ainsi que Tikz pour la gestion du graphique.

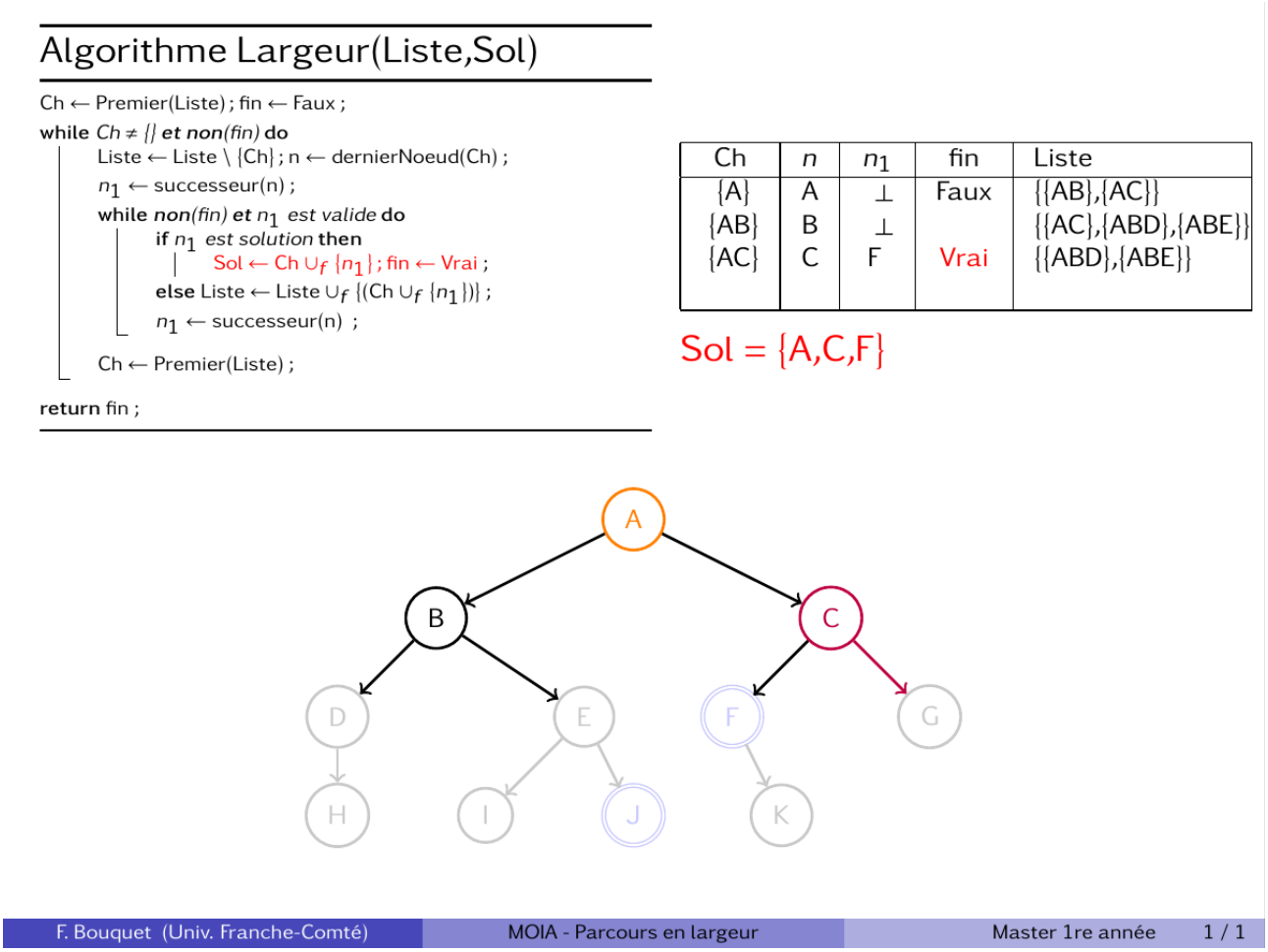

Illustration 1 - Exemple d'une diapositive

2 - Analyse des besoins

Une analyse des besoins est nécessaire à la compréhension du projet. Elle permet de définir ce qui est essentiel au projet et ce qui est optionnel ce qui nous a aidé à définir dans quel ordre réaliser les différentes fonctionnalité de notre logiciel.

2.1 - Besoins de l'utilisateur

L'utilisateur a besoin de définir son algorithme par son pseudo code en montrant à quel déroulement de l'algorithme chaque diapositive correspond. Donc le logiciel doit disposer d'un éditeur pour le pseudo code ainsi qu'une mise en évidence sur ce dernier.

L’utilisateur a besoin de montrer clairement les valeurs contenues dans les variables définies dans le pseudo code. Il doit également pouvoir montrer à quel moment une variable change de valeur. Donc le logiciel doit contenir un tableau représentant les variables du pseudo code ainsi qu’une mise en évidence des variables au moment d’un changement de valeur.

L’utilisateur a besoin d’éditer un graphique représentant le déroulement de l’algorithme pas à pas ainsi qu’une mise en évidence du chemin parcouru à un moment précis. Le logiciel doit donc comporter une partie permettant l’édition d’un graphe ainsi qu’une mise en évidence d’une partie précise du graphe représentant le chemin qui est parcouru.

L’utilisateur à également besoin de visualiser le rendu diapositive par diapositive. Il est donc impératif de pouvoir effectuer un changement de diapositive afin de visualiser l’avancement ou le rendu final d’un diaporama.

Pour finir le logiciel doit retranscrire les informations saisies par l’utilisateur sous la forme d’un code LaTeX.

2.2 - Aspect visuel

Pour une meilleur vision du rendu final d’un diaporama nous souhaitons que l’interface de notre logiciel soit le plus proche possible de la représentation d’une diapositive. Nous souhaitons également rendre l’aspect visuel du logiciel proche des logiciels d’édition couramment utilisés tel que les éditeurs de textes, de tableaux, etc.

Ainsi cinq parties seront présentes lors de l’édition d’un diaporama.

Dans un premier temps, un menu permettant l’ajout d’un nouveau diaporama, sa sauvegarde ainsi que la fermeture du logiciel. De même que L’ajout et la suppression des colonnes et lignes du tableau correspondant aux variables. Ainsi qu’une rubrique d’aide à la manipulation du logiciel.

La deuxième partie est constitué d’une barre d’outils repositionnable par l’utilisateur. Cette barre d’outils comprend les boutons contrôlant le changement de diapositive ainsi que l’édition du graphe représentant l’évolution de l’algorithme.

Nous en venons à la partie permettant l’édition du pseudo code. Nous souhaitons la faire ressembler à un éditeur de texte standard comprenant les numéros des lignes puis la mise en évidence d’une ligne par un rond correspondant à la ligne ainsi que le pseudo code.

Ensuite, dans la quatrième partie, nous retrouvons tableau permettant la

représentation des valeurs des différentes variables du pseudo code. Le changement de la valeur d'une variable d'une diapositive à une autre entraînant le mise en évidence de cette valeur.

Pour finir, la cinquième partie correspond au graphique représentant le déroulement de l'algorithme. Le chemin suivit durant le déroulement de l'algorithme est mis en évidence par le changement de couleur de la partie correspondante du graphe. Le parcours du graphe est visible par l'apparition des zones parcourus précédemment par l'algorithme. Les états finaux sont représenter par un double cercle et les états non accessible par un cercle barré.

2.3 - Aspect technologique

Pour la réalisation de ce projet nous avons choisi d'utiliser java. Ce langage étant le langage objet que nous avons le plus utilisé, nous avions également une matière lors de notre 5ème semestre qui nous apprend comment utiliser la bibliothèque graphique swing de java. Cette matière à pu nous aider pour la réalisation de se projet ainsi que le projet nous a aider pour l'élaboration du projet correspondant à cette matière.

L'implémentation de tests unitaire nous permettra de déceler d'éventuels erreur durant le développement de notre logiciel et nous aidera à améliorer la structure de ce dernier.

Afin de suivre pas à pas le développement de notre logiciel nous avons choisi d'utiliser le logiciel de gestion des versions Git que nous n'avions jamais utilisé auparavant.

3 – Début du projet

Nous avions deux choix pour effectuer ce projet, dans un premier temps reprendre le travail réaliser durant l'année 2015 – 2016 par un autre binôme ou recommencer le projet du début.

3.1 – Premier choix

Nous nous sommes d'abord tourné vers le premier choix, reprendre le travail laisser par un autre binôme. Le binôme précédant a choisi de réaliser le projet en C++ et Qt. Nous avons rencontrer de nombreuses difficultés tout d'abord avec la compréhension du code, nous n'avions fait que très peu de C++ et nous n'avions jamais utiliser Qt. L'absence de nombreux commentaires,

de la documentation et l'implémentation étrange à entraver grandement notre travail de compréhension du code.

Le projet de ce binôme était presque entièrement terminé il manquai seulement les tests unitaires.

Le faite d'avoir essayé de reprendre ce code nous à appris beaucoup de chose sur le code à laisser à d'éventuels personnes. Tout d'abord l'importance d'avoir un code clair et de ne pas oublier de commenter ce code tout au long du développement ainsi que l'importance de la documentation à apporter à ce code.

N'étant pas suffisamment à l'aise avec le langage et n'ayant pas beaucoup avancer dans la compréhension du code pendant une période d'un mois nous nous sommes tourné vers la deuxième solution.

3.2 – Second choix

---

4 – Conception

4.1 - Structure du logiciel

Pour ce projet nous avons choisi d'utiliser l'architecture Modèle-Vue-Contrôleur pour l'implémentation des classes. DIAGRAMME UML

4.2 - Conception des différents éléments

4.2.1 - Changement de diapositive

Le changement de diapositive s'effectue à l'aide de boutons présent dans la barre d'outils, on peu également changer à une diapositive précise en indiquant son numéro. Il est également possible de ne changer qu'une seule partie ou plusieurs, éditeur du pseudo code, le tableau des variables ou le graphique, en sélectionnant cette ou ces parties à l'aide de checkboxs.

4.2.2 - Editeur du pseudo code