Année universitaire 2016-2017
Université de Caen Basse-Normandie

Rapport sur le premier semestre

Alexis Carreau
Thomas Lécluse
Emma Mauger
Théo Sarrazin
L2 Informatique

Table des matières

1 Introduction **2**
1.1 Présentation 2
1.2 Objectifs 2
1.3 Déroulement et planning 2

2 Détails techniques **2**
2.1 Bibliothèques utilisées 2
2.2 Techniques employées 3

3 Structure générale **3**
3.1 Répartition en modules 3
3.2 Différentes tâches 3

4 Conclusion **3**
4.1 Améliorations 3

1 Introduction

1.1 Présentation

Nous avons choisi de réaliser l'IDE (Integrated Developpement Environnement), car nous voulions créer un outil que nous pourrions utiliser par la suite. Ce sujet nous semblais donc intéressant à faire.

Un IDE fournit des facilités au programmeur pour le développement logiciel. Il a pour but de maximiser la productivité du programmeur. Il contient généralement :

— un éditeur de texte,
— un interpréteur,
— un debugger,
— un compilateur,
— des options avancées comme la recherche de termes, l'autocomplétion, la coloration syntaxique...

1.2 Objectifs

Nous devons d'ici à la fin de l'année réaliser un IDE qui contiendra les éléments cités ci-dessus. Pour la coloration lexicale et l'analyse syntaxique du code nous utiliserons les programmes Lex et Yacc.

1.3 Déroulement et planning

Nous avons réparti équitablement le travail et le temps de travail de chacun. Malgré le fait que deux membres du groupe (Thomas et Théo) connaissaient déjà la bibliothèque graphique que l'on utilise et partaient donc avec un avantage non-négligeable.
Les deux autres membres du groupe ont donc été aidés au début afin de leur expliquer les bases et de pouvoir démarrer sans trop de problèmes.

Nous avons réparti les différentes tâches à effectuer pour le projet entre les membres du groupe. Certaines sont plus longues ou plus complexes, c'est pourquoi certains membres du groupe on une liste moins remplie que d'autres. Il ne faut donc pas se fier uniquement à cette liste.

— Alexis : navigation / gestion des projets
— Emma : menu / Architecture / recherche bibliographique
— Théo : Coloration syntaxique / Analyse lexicale / Interface
— Thomas : Documentation / Interface / style texte
— Rapports et présentation : tout le monde

2 Détails techniques

2.1 Bibliothèques utilisées

Nous avons choisi la bibliothèque graphique QT (version 4.8.7, prévue à la base pour le C++) ce pourquoi nous avons aussi eu besoin de Pyside (version 1.2.4), qui fait le lien vers le langage Python (version 3.4).

La raison de ce choix est le fait que QT est un outil plus puissant de par sa richesse de fonctionnalités. Sa documentation est de plus très fournie car les utilisateurs de QT sont plus nombreux que ceux de Tkinter.

2.2 Techniques employées

Nous avons choisi un style de programmation orienté objet, puisque plus logique dans un projet de cette envergure. L'objet nous permet de mettre en application les notions vues en cours. Cela nous permet de personnaliser en ré-implémentant des classes de QT pour les adapter à nos besoins.

Lex est un programme qui permet de reconaître des tokens qui sont des éléments d'une chaîne de caractères. Cela à l'aide de règles de grammaires que l'on lui passe en entrée.
Dans notre cas, nous utilisons Lex pour reconnaître les éléments que l'on écrit, afin de colorier ces derniers en fonction de leur rôle (identifiant, entier, déclaration...). On utilise également la sortie générée que l'on passe à un autre programme : Yacc.

Yacc permet de générer un arbre syntaxique abstrait qui nous permet de vérifier que la syntaxe de notre code est correcte. Aussi, cela ne permet de proposer une complétion automatique en fonction de ce que l'on écrit.

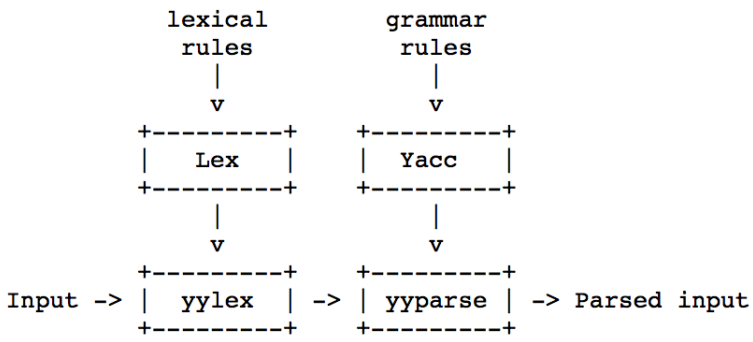

FIGURE 1 – Shéma résumant le foncionnement de Lex et Yacc.

3 Structure générale

3.1 Répartition en modules

3.2 Différentes tâches

ON EXPLIQUE CHAQUE PARTIE DU CODE

4 Conclusion

4.1 Améliorations