Dossier de conception

REV party

GAUTHIER Axel

Cadre administratif :

- Nom : Gauthier
- Prénom : Axel
- Groupe de TP : TPA61
- Adresse mail : axel.gauthier@univ-tlse3.fr
- Code APOGEE : EDFIN3G1
- Programme : RVE party
- Année : 2018
- Tuteur : NOUGNANKE KOKOUVI Benoit
- Adresse mail tuteur : benoitnougnanke@gmail.com

Cadre de développement :

Principalement deux machines seront utilisées pour la réalisation de ce projet. Un ordinateur portable moyennement performant sous les systèmes Debian et Windows 10. Un ordinateur fixe très performant sous Windows 10 avec un accès SSH.

Les deux machines possèdent les logiciels nécessaires à la réalisation du programme (compilateur, IDE, outils de versionning, etc.) excepté Doxygen.

Pour gérer les sauvegardes je vais utiliser l'outils git avec comme gestionnaire de répertoire GitHub.

Les logiciels indispensables à la conception du programme sont déjà présents sur les machines.

Projet :

Il y a deux groupes de structures de données importants. Les matrices dynamiques ainsi que les graphes orientés. Le premier sert à stocker les données provenant du fichier CSV. Le deuxième sert pour la gestion des données avec l'algorithme Condorcet. Il faudra traiter des entiers et des caractères.

Il y aura environ 13 fichiers de code. Les différents rôles seront : gestion du fichier CSV, gestion de l'algorithme Condorcet avec ses dépendances, gestion de l'algorithme Uninomial, gestion des structures de données, gestion des options de lancement, affichage, communication avec le fichier python et gestion du programme principal. Il faudra compter en plus le fichier python. Les structures de données partagées seront les matrices et les graphes orientés. Elles seront transmises par passage par référence.

Il y aura environ 10 headers. Les différents rôles seront gestion du fichier CSV, gestion de l'algorithme Condorcet et ses dépendances, gestion de l'algorithme Uninomial, gestion des structures de données et gestion des options de lancement, affichage et communication avec le fichier Python.

L'utilisation d'un Makefile est nécessaire.

Je pense procéder ainsi : création des structures, traitement du fichier CSV, gestion des options, algorithmes uninomiaux, algorithme Condorcet, communication avec le fichier python, affichage et mise en place du fichier principal. Les différents fichiers gérant certains cas de l'algorithme Condorcet dépendent de ce dernier. L'affichage dépend du retour des différents algorithmes de scrutin. Ces derniers dépendent de la création du fichier CSV. Et pour finir, sans structure de données, rien ne fonctionne.

La liste des fonctionnalités avec leurs paramètres :

- lecture_csv : chemin du fichier CSV
- struct_donnees : -
- algorithmes uninomiaux : données à traiter sous forme de matrice de chaine de caractères
- condorcert : données à traiter sous forme de matrice de chaine de caractères
- communication_python : graphe orienté sous forme de liste
- gestion_options : options représentées par une chaine de caractères
- affichage : liste d'entiers
- main : tableau de chaines de caractères

Aucune variable globale ne devrait être utilisée.

Gestion du projet :

Les étapes principales semblent être dans l'ordre temporelle : le traitement du fichier CSV, gestion des options, les algorithmes uninomiaux, l'algorithme condorcet, affichage du graphe et mise en place du programme principal.

Une estimation pour chaque étape :

- Traitement fichier CSV : 6h
- Gestion des options : 6h
- Algorithme scrutin uninomial : 3h
- Algorithme condorcet : 12h
- Affichage du graphe : 3h
- Mise en place du programme principal : 12h

Les tests dépendent de la globalité du code étudiés. Les boucles servant pour les variantes de l'algorithme condorcet seront vérifiées grâce à des invariants. Les fonctions seront vérifiées par des assertions et des tests unitaires. Enfin la cohérence du tout sera supervisé par des tests d'intégration.

Certains des tests sont déjà connus :

- Structures de données : un fichier rempli d'assertions vérifiant le bon fonctionnement de chacune des fonctions concernant les différentes structures de données.
- Algorithmes : les résultats qu'ils donneront seront comparés avec des résultats calculés de façon fiable. Des assertions seront donc utilisées.
- Traitement du fichier CSV : une liste de différents fichiers seront envoyés dans la fonction et ce qui en sortira sera comparé avec des fichiers traités « à la main ».

Pour le reste certains modules peuvent être testés à l'œil (vérifier que l'affichage est de la forme attendue par exemple). Il en reste également dont je ne sais pas encore comment les vérifier (communication avec le fichier python).

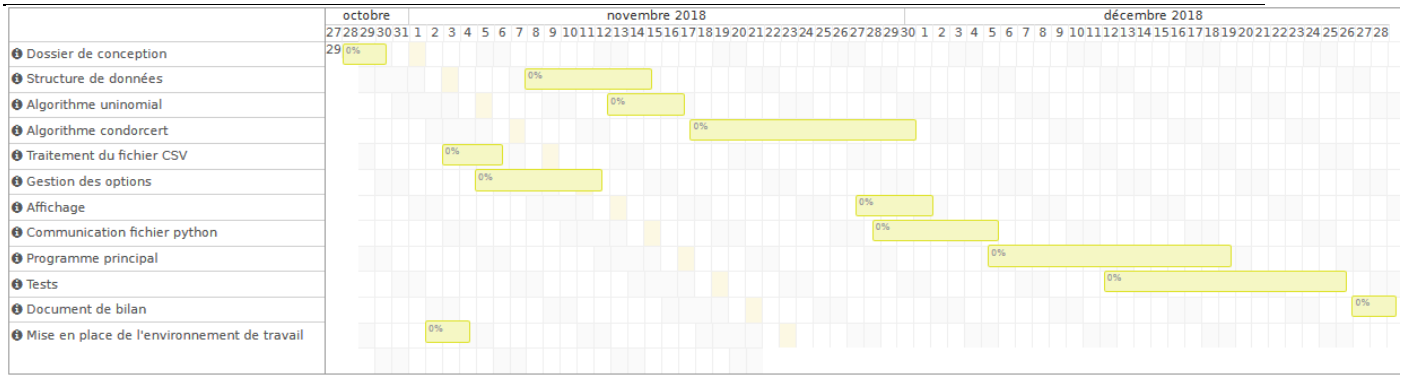
L'élimination de certains bugs se fera avec l'affichage de messages d'erreurs ou alors directement avec l'outils GDB.

Je prévois environ 18h pour les tests.

Oui ceci est indiqué sur le diagramme de gant.

Je vais tenter de coller au plus proche du temps que je me suis accordé. Si du retard se manifeste, la première solution sera de trouver du temps supplémentaire pour finir tout le projet avant la date limite. Si jamais il y a un retard majeur, certaines fonctionnalités seront carrément abandonnées.

Gantt :


octobre
novembre 2018
décembre 2018
27 28 29 30 31 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28
Dossier de conception
Structure de données
Algorithme uninomial
Algorithme condorcert
Traitement du fichier CSV
Gestion des options
Affichage
Communication fichier python
Programme principal
Tests
Document de bilan
Mise en place de l'environnement de travail
0%