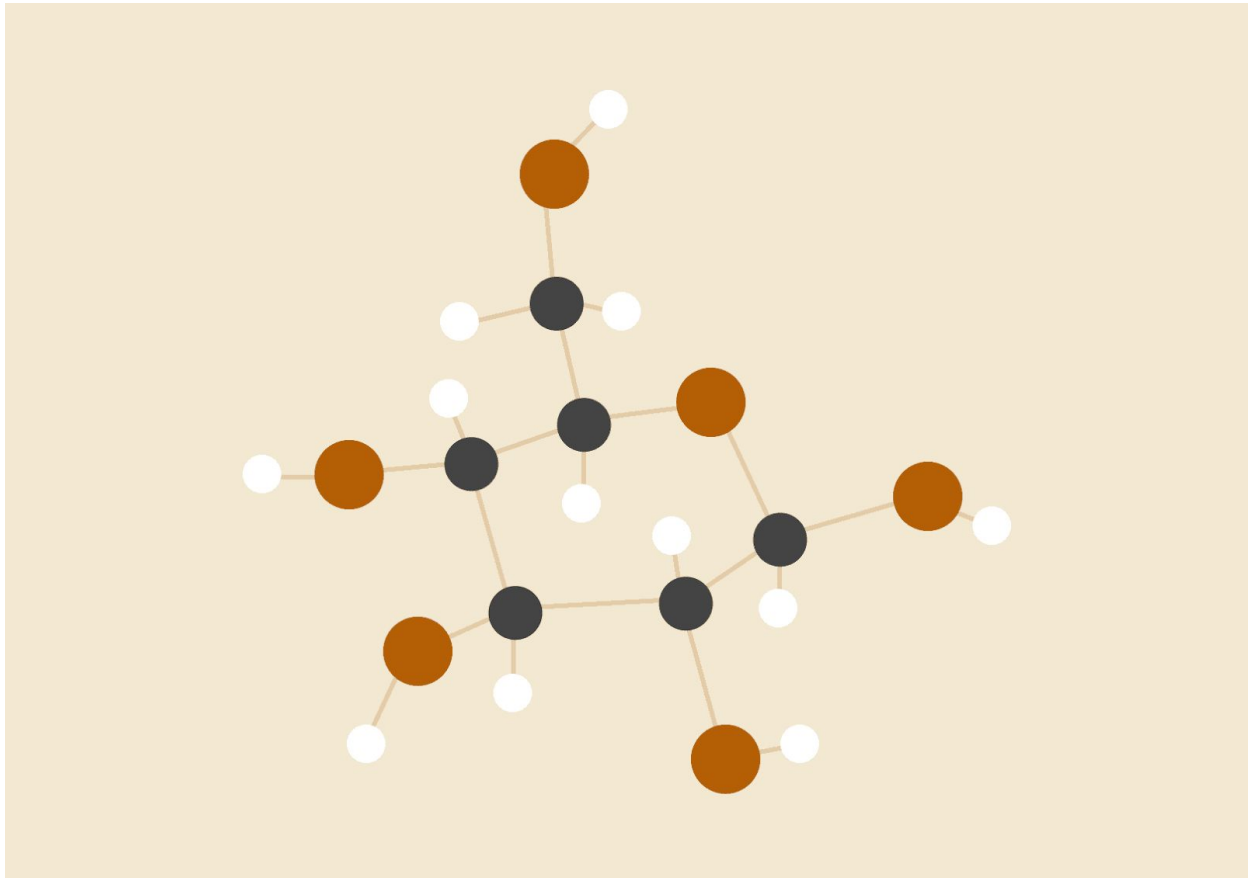


RAPPORT PROJET

PERCEPTRON MULTI-COUCHE

GAUDILLAT Eliott - BERNAT Loan



Informatique S3

19/12/2020

CUPGE - A31

1- Introduction

2- Utilisation de Git

3- Ecriture des modules

a) Partie GAUDILLAT

b) Partie BERNAT

4- Compilation et Test

5- Ressenti

INTRODUCTION

Pour la réalisation de ce projet, nous avons décidé de travailler simplement en binôme car nous travaillons régulièrement ensemble sur de nombreux projets extra-scolaires en informatique ou sur d'autres thèmes. Mr Gaudillat s'est occupé du module *utils.c* et *matrices_operation.c* . Tandis que, Mr Bernat, s'est occupé du module *matrice.c* , *matrice_accesseur.c* et de la rédaction de ce dossier.

UTILISATION DE GIT

L'utilisation de Git n'a pas posé tellement de problèmes. À chaque nouveau module que l'on commence, nous utilisons une nouvelle branche pour éviter de "polluer" la branche main. Nous sauvegardons assez régulièrement nos progrès afin de nous aider mutuellement dans nos problèmes. Une fois la rédaction du module terminée, nous fusionnons notre branche de travail avec le *main* afin de pouvoir partager notre travail.

ÉCRITURE DES MODULES

1. Partie GAUDILLAT

Le module *utils.c* est basé sur la fonction `lire_donnee`. En effet les autres fonctions du modules sont juste des réutilisations de `lire_donnee` pour les différents types. Au final, le plus compliqué dans ce module fut la compression de la fonction en elle-même et du fonctionnement de la boucle en son sein.

Enfin , dernière petite remarque sur ce premier module pour la fonction `lire_entier_intervalle` , j'ai considéré l'intervalle donné correct, c'est-à-dire que le min et bien inférieur au max.

Pour ce qui est du module *matrice_operations.c* les fonctions demandées étaient claires et sans réelles difficultés car il s'agissait plutôt d'applications mathématiques pures. Cela revenez à écrire les formules générales de calcul de matrices.

2. Partie BERNAT

J'ai commencé personnellement par le module *matrice.c* où j'ai réalisé les fonctions dans l'ordre où elles m'étaient demandées. Je pense que l'unique difficulté du module était de saisir les petites particularités et de bien différencier les pointeurs des adresses. Je me suis inspiré de votre fonction afin de rapprocher ma façon de coder de celle ci.

```
int matrice_ligne ( struct matrice_s * * m , float * t , int n )
{
    int i ;
    if( *m == NULL )
    {
        *m = creation_matrice ( 1 , n ) ;
    }
    for ( i = 0 ; i < n ; i++ )
        (*m)->matrice[0][i] = t[i] ;
    return 0 ;
}
```

La seule particularité du module se trouve dans la fonction de tirage aléatoire. Pour réaliser ce tirage entre -1 et 1. J'ai utilisé cette ligne qui effectue deux tirages entre 0 et 1 et qui soustrait les deux pour avoir un résultat au minimum à -1 (Premier tirage 0 et deuxième 1) et au maximum à 1 (Premier tirage à 1 et deuxième à 0).

```
Nb= (float)rand() / (float)RAND_MAX - (float)rand()/(float)RAND_MAX;
```

Pour les fonctions qui lisent ou écrivent dans un fichier, j'ai supposé qu'ils étaient déjà ouverts en écriture ou en lecture ou en écriture et lecture. Mais j'ai constamment vérifié que le pointeur renvoyait bien sur un fichier avant de poursuivre la suite de la fonction

```
if ( f == NULL) //On verifie que le fichier a bien pu etre ouvert
{
    printf("ERREUR Impossible de lire le fichier");
}
```

Enfin, pour écrire et lire dans le fichier j'ai utilisé les fonctions *fprintf* et *fscanf* !

Le module *matrice_accesseur*, quant à lui, fut une formalité et ne posa pas de problèmes particuliers.

J'ai essayé cette dernière semaine de m'attaquer au module *activation* mais sans succès car je ne comprenais pas ce qu'il était demandé.

COMPILATION ET TEST

Malheureusement, nous avons essayé de compiler à plusieurs reprises afin de tester nos programmes, mais nous avons toujours la même erreur récurrente, nous indiquant que certaines fonctions étaient déjà définies dans vos programmes, nous empêchant d'avancer.

```
vocstartsoft:~/projet/Projet_EG_LB (main) $ make test
gcc -Wall -g3 -Wno-unused-value -I include -l lib src/main_apprentissage.c -o bin/apprentissage lib/specification.o lib/couches_sortie.o lib/couches_activation.o lib/couches_matrice.o lib/activation.o lib/matrices_accesseur.o -lm
lib/activation.o: In function `demande_fonction_d_activation':
/home/ec2-user/projet/Projet_EG_LB/src/activation.c:56: multiple definition of `demande_fonction_d_activation'
lib/specification.o:/home/chevalie/src/c/PMC/src/specification.c:82: first defined here
lib/matrices.o: In function `copie_matrice':
/home/ec2-user/projet/Projet_EG_LB/src/matrices.c:170: multiple definition of `copie_matrice'
lib/matrices_operations.o:/home/chevalie/src/c/PMC/src/matrices_operations.c:11: first defined here
collect2: error: ld returned 1 exit status
make: *** [bin/apprentissage] Error 1
```

Nous n'avons pas réussi à comprendre la source de cette erreur malgré nos efforts. Cela vient peut être d'un problème dans les .h ?

RESSENTI

Nous avons personnellement trouvé le projet peut être un peu trop ambitieux pour être réalisé par des étudiants simplement après 3-4 mois de manipulation de C. Cependant il en reste intéressant et il pourrait être utile de s'y repencher dessus durant le S4 pour voir ça sous un œil un peu plus aguerri. Le problème de compilation est un vrai frein pour nous car nous ne pouvons pas voir réellement le fonctionnement du projet et son but. C'est difficile d'avancer quand nous ne connaissons pas le point de départ et la ligne d'arrivée. Malgré le fait que nous étudions relativement bien guidé, c'est une méthode de travail très "tâtonnante". La principale difficulté était de comprendre votre demande autour de ce projet.