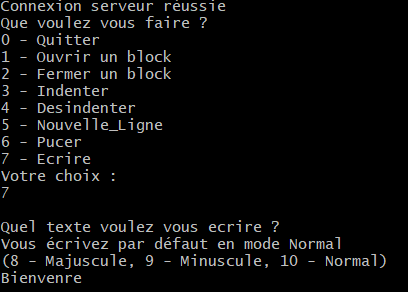
**Système et réseaux**

Membre :

* Julian Wicke
* Eliott Garnier****

## **But du projet**

Nous allons créer deux programmes qui permettent de saisir du texte enrichi via un réseau et d'envoyer ces informations au serveur, où elles seront enregistrées dans un fichier. Ce projet vise à démontrer la faisabilité de cette solution et sur sa capacité à gérer des applications à distance et à communiquer via un réseau.

Pour ce projet, nous allons travailler avec du texte enrichi qui a une largeur de 80 caractères. Nous allons fournir des outils pour écrire des mots dans ce texte. Si un mot dépasse la longueur de la ligne, il sera placé sur une nouvelle ligne. Il sera possible d'écrire les mots de trois façons différentes :

* NORMAL : la casse des caractères respecte celle du mot écrit ;
* MAJUSCULE : toutes les lettres du mot sont écrites en majuscule indépendamment de la casse originelle ;
* MINUSCULE : toutes les lettres du mot sont écrites en minuscule indépendamment de la casse originelle ;

Il sera possible de changer de mode d'écriture à tout moment pendant la rédaction du texte. Il sera également possible d'ouvrir des blocs de texte en les encadrant avec des symboles '+', '-' et '|'.

On pourra aussi créer des boîtes de texte récursivement, ainsi que d'ajouter ou de retirer des retraits à droite au début de chaque ligne. Les utilisateurs pourront également ajouter des puces dans le texte en fonction du niveau de retrait. Il sera également possible d'indenter des boîtes de texte et de retraiter le texte à l'intérieur de ces boîtes.

## **Fonctionnement**

Il y aura 2 programmes :

* Un programme serveur :
  + Le serveur est un programme prévu pour tourner indéfiniment et pouvant accepter la connexion d’un client.
  + Lors de la connexion, le serveur crée un fichier pour associer au client.
  + Les fichiers pourront être numérotés en fonction de l’ordre d’arrivée des clients.
  + Recevoir des ordres de la part des clients
* Un programme client :
  + Le client en se connectant aura un message qui s’affiche lui indiquant les actions qu’il peut réaliser.
  + Les fonctionnalités disponibles :
* L’écriture d’un mot ;
* L’ouverture d’un bloc ;
* La fermeture d’un bloc ;
* L’ajout d’une indentation ;
* Le retrait d’une indentation ;
* L’ajout d’une puce ;
* Le retour à la ligne ;
* Et bien sûr la fermeture du fichier au moment où le client quitte le service.
  + Distinguer les ordres fournis par le client pour l’associer à son fichier de texte enrichi : C’est-à-dire les deux clients doivent être capables d’ajouter des éléments à leurs fichiers respectifs sans que ceux-ci ne se mélangent.
* Le client qui tourne tant que l’utilisateur n’indique pas son désir de quitter l’application et il doit pouvoir :
  + Connecter un serveur à travers le réseau ;
  + Récupérer une succession d’ordres d’écriture fournis par l’utilisateur ;
  + Transmettre l’ordre au serveur ;

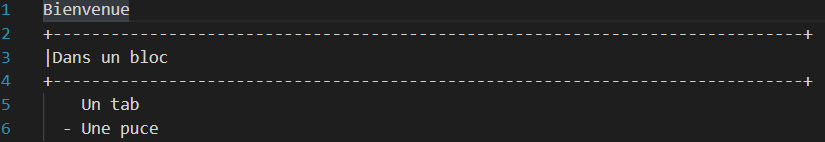
Nous retrouverons aussi un makefile. Ce fichier contient les informations nécessaires pour compiler vos programmes en utilisant les options de compilation '-Wall' et '-Wextra'. Votre programme doit être correctement compilé et ne doit pas générer de warnings.

## **Bilan**

Le projet pour nous n’est pas terminé. Avec plus de temps, nous aurions aimé améliorer l’affichage du côté client pour avoir un aspect visuel plus facilement lisible.

De plus certaines fonctionnalités sont manquante comme par exemple celle qui permet l’écriture “|” en fin de ligne dans un bloc ou aussi le passage à la ligne automatique si le client écrit une phrase qui dépasse la capacité du message.

Sinon nous sommes fiers de notre projet car nous avons réussi à réaliser la majeure partie de ce qu’on voulait faire. De plus, ce projet nous a permis de mettre en pratique ce qu’on a vu en cours et d’améliorer nos compétences et notre travail en équipe.

****

