

# PROJET FAR

## UNE APPLICATION DE MESSAGERIE INSTANTANÉE



POLYTECH  
MONTPELLIER



IG

AYOUB MOUJANE

PERRIN PIERRE

WIEGANDT JULIEN



# SPRINT 3

## FEATURES IMPLÉMENTÉES

- Fonctionnalité : Transfert de fichiers
  - un client a la possibilité de :
  - Lister les fichiers du répertoire '/filetosend', qu'il peut envoyer
  - Envoyer un fichier provenant du répertoire '/filetosend'
  - Recevoir des fichiers dans un répertoire '/filereceived'

# PROTOCOLE DE COMMUNICATION

- Nous avons décidé de mettre en place un protocole de communication en mode connecté (TCP)
- Pour envoyer un fichier un client envoi « file » dans le chat, à son écran s'affiche les fichiers présents dans son dossier **/filetosend**,
- Il entre ensuite le nom d'un fichier valide
- Le client « passe en mode serveur », il attend la connexion d'autre clients pour transférer le fichier choisi
- Le serveur reçoit l'information : le client « envoyeur » est prêt à partager des fichiers. Le serveur transmet aux autres clients l'adresse IP et le port du client « envoyeur ».
- Les clients recevant les données d'un client « envoyeur » se connectent automatiquement à ce dernier et sont prêt à recevoir le fichier.

# CHOIX DE CONCURRENCE ET DE SYNCHRONISATION

- Lorsqu'un client se connecte au serveur, ce dernier créer un thread client pour que le client puisse entrer son pseudo et communiquer sans bloquer les autres.
- Lorsqu'un client se connecte à un autre client lors de la phase d'envoi de fichier, il va créer un thread d'envoi du fichier au client qui vient de se connecter. Cela permet de pouvoir envoyer des fichiers en parallèles.
- Mise en place d'un mutex pour gérer une zone critique lors de la connexion du client au serveur.

# ORGANISATION DU CODE

## OUTIL DE DÉVELOPPEMENT

### RÔLE DE CHACUN

- Organisation du code : utilisation de fonctions pour éclaircir au maximum avec un main par programme
- Outils de développement utilisés :
  - Messenger pour la répartition des taches et le suivi de l'avancement
  - Discord pour le réglage des problèmes rencontrés
  - Visual Studio Code et SublimeText pour coder
- Rôle de chacun :

|                 | Création serveur peer | Transmission des informations peer | Réception fichier | Envoi fichier | Diagramme séquence |
|-----------------|-----------------------|------------------------------------|-------------------|---------------|--------------------|
| Ayoub Moujane   |                       |                                    |                   |               |                    |
| Pierre Perrin   |                       |                                    |                   |               |                    |
| Julien Wiegandt |                       |                                    |                   |               |                    |

# DIFFICULTÉS RENCONTRÉES

- Difficultés :
  - Considérations sur les formats des données
  - L'utilisation de données formatées
  - La problématique de la création du serveur peer
  - Problématique liée à la réception des informations dans le thread lié au fait que les variables utilisées sont des pointeurs
  
- Pistes d'améliorations :
  - Faire en sorte d'afficher dans un terminal à part les fichiers à envoyer

# DEMONSTRATION

## SPRINT3

# SPRINT 4

## FEATURES IMPLÉMENTÉES

- Sprint 4 : gestion de chaînes de discussion
  - un client a la possibilité de :
  - Discuter dans un salon de discussion avec d'autre clients
  - Lister les salons existants avant de s'y connecter
  - Création/suppression/modification de salon avant de se connecter

## PROTOCOLE DE COMMUNICATION

- Nous sommes restés sur un protocole de communication en mode connecté (TCP)
- Lorsqu'un client se connecte celui-ci doit choisir un salon avant de pouvoir discuter.
- Il entre un numéro de salon valable pour s'y connecter.
- Commande '/list-channels' pour lister les salons
- Commande '/create-channel' pour créer un salon
- Commande '/delete-channel' pour supprimer un salon
- Commande '/modify-channel' pour modifier un salon
- Le protocole d'échanges de messages et fichiers ne changent pas par rapport au sprint 3.

# CHOIX DE CONCURRENCE ET DE SYNCHRONISATION

- Un thread pour la connexion à un salon
- Un thread pour la gestion des salons
- Une fois entré dans un salon un thread dédié au client est initié. Il sert à recevoir les messages et les transmettre aux membres du salon dans lequel le client se trouve.

# ORGANISATION DU CODE

## OUTIL DE DÉVELOPPEMENT

### RÔLE DE CHACUN

- Organisation du code : utilisation de fonctions, de structure de données dans des librairies séparées .h
- Outils de développement utilisés :
  - Messenger pour la répartition des taches et le suivi de l'avancement
  - Discord pour le réglage des problèmes rencontrés
  - Visual Studio Code et SublimeText pour coder
- Rôle de chacun :

|                 | Création de la structure de donnée salon | Création du thread de connexion à un salon | Gestion des input lors de la connexion au salon | Création des salons disponibles | Diagramme séquence |
|-----------------|--|--|---|---------------------------------|--------------------|
| Ayoub Moujane   |  |  |   |                                 |                    |
| Pierre Perrin   |  |  |   |                                 |                    |
| Julien Wiegandt |  |  |   |                                 |                    |

# DIFFICULTÉS RENCONTRÉES

- Difficultés :
  - Le client veut envoyer un fichier avant d'avoir rejoint un salon
  - Le parcours d'un tableau dont la longueur varie
  - Si un client ne choisit pas de salon il bloque les autres clients
  
- Pistes d'améliorations :
  - Mieux gérer la gestion des salons notamment la suppression.
  - Ajouter une fonctionnalité pour la déconnexion d'un salon suivi d'un retour vers la sélection des salons

# DEMONSTRATION

## SPRINT4