NetworkListener - Instancie la connexion du côté serveur **NetworkConnection** # rendre cette classe thread safe - Représente la connexion

TPC/Internet entre 2 machines

```
IPC (singleton)
# file de struct Message
# memoire partagée...
# ens de semaphore
+ addMessage(struct Message)
--> ajoute dans memoire partagee
--> si memoire partagée pleine
---> virer la dernière entrée
--> semaphore += 1 ### A vérifier
- constructeur qui gère sa mise en place
- gérer I/O de messages IPC
vue sur tous les proc fils (diffusion)
```

```
ProtocolCommands (DEFINE)
enum CommandType {
 Message
struct Command {
 CommandType type;
struct CommandMessage {
 CommandType type;
 timestamp
 pseudo
 contenu
```

Client

- NetworkConnection nConnection
- pushToServer(string msg)
- --> str = ProtocolEncoder.encodeMessage(msg)
- --> nConnection.send(str)
- pullFromServer()
- --> Command c = nConnection.fetchDataFromSocket()
- --> if (c.

Interface

- clear()
- write()

Server

ClientManager

```
[Boucle bloquante select] avec un NetworkListener et écoute clavier
--> NetworkListener: on a un NetworkConnection
---> Fork
----Fils--> Crée un nouveau Client(n)
----Fils--> BREAK
--> Ecoute clavier : coupe le serveur proprement
```

Client

vars

- NetworkConnection nConnection

methods

- pushToClient()
- --> utiliser NetworkConnection pour envoyer les données
- pullFromClient(struct Message m)
- --> IPC.getSharedInstance().addMessage(m)
- ThreadPush()
- --> while (bloquant P1)
- ---> pushToClient()
- ThreadPull()
- --> while listen bloquant NetworkConnection -> m
- ---> pullFromClient(m)
- + constructeur(NetworkConnection)
- --> pushToClient()
- --> créer ThreadPush
- --> créer ThreadPull