Lucas MOULIN CIN2  
Malo BARBAUD

**PROJET PIXEL WAR:**

**Architecture du projet :**

1. ***Introduction***

Le projet "PixelWar" consiste à mettre en place une architecture serveur capable de dialoguer avec plusieurs clients simultanément grâce à l'appel système poll() qui surveille les descripteurs de fichiers. Le dialogue se fera via une socket de communication en protocole serveur TCP (port à spécifier en ligne de commande, option -p).

1. ***Fonctionnalités du serveur***

Le serveur doit être capable de :

* Gérer l'affichage des messages bruts envoyés par les clients
* Faire la gestion des déconnexions en supprimant le client de la liste chaînée
* Attendre la connexion de plusieurs clients simultanément grâce à l'appel système poll()
* Stocker les clients connectés dans une liste chaînée
* Répondre aux commandes envoyées par les clients

1. ***Fonctionnalités des clients***

Les clients pourront envoyer des commandes au serveur en lui demandant des informations ou en lui demandant d'initialiser la matrice avec des pixels de couleur RGB choisis par le client.

1. ***Liste des commandes possibles***

Les clients pourront envoyer les commandes suivantes au serveur :

* /getMatrix : Retourne la matrice des pixels en RBG 8 bits codés BASE64 en partant du haut à gauche et ligne à ligne sans séparateur
* /getSize : Retourne la taille de la matrice (LxH) en ASCII\_INT x ASCII\_INT
* /getLimits : Retourne le nombre maximum de pixels pouvant être modifiés par minute et par adresse IP source en ASCII\_INT
* /getVersion : Retourne la version du protocole en ASCII\_INT
* /getWaitTime : Retourne le nombre de secondes à attendre avant de pouvoir envoyer un nouveau pixel (0 si la limite n'est pas atteinte) en ASCII\_INT
* /setPixel ASCII\_INTxASCII\_INT ASCII\_COLOR\_B64 : Modifie un pixel en position (x,y) avec la couleur en RBG 8 bits codée BASE64 spécifiée par le client. Le serveur renvoie un code de retour (voir ci-dessous).

Le serveur doit donc être capable de décoder ces commandes et de renvoyer les informations demandées ou de modifier la matrice en fonction de la commande reçue.

1. ***Codes de retour***

Le serveur renverra les codes de retour suivants :

* 00 OK
* 10 Bad Command
* 11 Out Of Bound
* 12 Bad Color
* 20 Out Of Quota
* 99 Unknown Command

1. ***Options et limites***

Le serveur prendra en compte les options et limites suivantes :

* PORT : port d'écoute du serveur. La valeur par défaut est 5000, mais il est possible de spécifier un autre port en ligne de commande avec l'option "-p". Si ce port est inférieur à 1024, les droits root seront nécessaires.
* L : Largeur de la matrice. Par défaut, elle sera fixée à 80.
* H : Hauteur de la matrice. Par défaut, elle sera fixée à 40.
* RATE\_LIMIT : Nombre maximum de modifications de pixels par minute et par adresse IP source. Par défaut, il sera fixé à 10.

1. ***Fonctions***

La fonction "ajouter\_client" qui sera appelée lorsqu'un nouveau client se connecte au serveur. Cette fonction devra ajouter le client à la liste chaînée des clients connectés et surveiller sa socket avec poll().

La fonction "message" qui sera appelée lorsqu'un client envoie un message au serveur. Cette fonction devra décoder la commande et modifier la matrice ou renvoyer les informations demandées en fonction de la commande reçue.

La fonction "supprimer\_client" qui sera appelée lorsqu'un client se déconnecte du serveur. Cette fonction devra supprimer le client de la liste chaînée des clients connectés.

La fonction "main" qui mettra en place le serveur en surveillant la socket d'écoute et les sockets des clients avec poll() et qui appellera les fonctions "ajouter\_client", "message" et "supprimer\_client" en fonction des événements.