

EXERCICE SOCKET

En équipe de 3 ou 4.

MATÉRIEL REQUIS

- 3 Raspberry PI 400
- Routeur du B536

POUR ALLER PLUS LOIN

- Mini-Tank

SOMMAIRE

- Création de l'application Client
- Création de l'application Serveur

Vous devez créer une application de chat en temps réel. Le serveur est celui qui reçoit les connexions par socket et qui reçoit également l'ensemble des messages. Suite à la réception des messages, le serveur doit envoyer le message à tous les clients à l'exception de l'expéditeur.

CRÉATION DU SERVEUR

La partie serveur est de loin la partie la plus complexe de cet exercice. Puisque plusieurs clients peuvent être connectés en même temps l'utilisation des threads est sans doute nécessaire. L'envoi de donnée par socket nécessite d'envoyer des tableaux d'octet. Vous devrez donc transformer vos chaînes de caractères(string) avec `.encode('utf-8')`. **Vous devez choisir un port et l'ouvrir.** Dans un contexte réel, on devrait porter attention à : https://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers. Par principe, évitez les ports inférieurs à 5001.

Le serveur doit :

- Autoriser la connexion d'au moins 3 clients
- Recevoir les messages des clients
- Envoyer les messages aux clients à l'exception de l'expéditeur

Au final, vous devriez avoir 1 raspberry PI en serveur avec 2 raspberry PI client qui s'y connecte pour s'échanger des messages.

CRÉATION DU CODE CLIENT

Le client doit :

- Se connecter automatiquement au serveur
 - L'IP et le port peuvent être définis en constante au début du script
- Prendre les inputs de l'utilisateur dans le terminal
- Envoyer les messages de l'utilisateur au serveur
- Afficher dans le terminal les messages reçus du serveur
 - L'affichage doit être le IP du client suivi du message.

SI VOUS TERMINEZ, À L'AVANCE

ENVOIE D'IMAGE

En utilisant les sockets, tentez d'envoyer une image et de la recevoir de l'autre côté et de l'afficher à l'écran.

MINI-TANK

Tentez votre chance avec le mini-tank. Vous devez être connecté à son réseau. Vous devez envoyer un json avec la commande. Le code est fourni qui y roule est fournis pour vous aider. Exemple :

```
{  
  cmd:"buzz"  
}
```

Également, un heartbeat doit être transmit à interval régulier sinon celui-ci va considérer la connexion perdue.

```
{  
  cmd:"heartbeat"  
}
```

```
{  
  cmd:"robotMovement",  
  angle :10  
  strength 50:  
}
```

La force doit être entre 0 et 100