

1 TP3 : TCP

Le but de ce TP est d'implémenter le jeu mastermind en réseau. Dans ce jeu, un joueur (le *codeur*) choisit une combinaison secrète de 4 pions colorés. Le but de l'autre joueur (le *décodeur*) est de deviner la combinaison secrète. Pour cela, il propose des combinaisons de pions et le codeur indique :

1. le nombre de pions dont la position et la couleur correspondent avec la combinaison secrète
2. le nombre de pions dont seule la couleur correspond (ce sont donc les pions de la bonne couleur mais incorrectement positionnés.)

1.1 Fonctionnement général

Le serveur jouera le rôle du codeur et le client celui du décodeur.

- Lors de la connexion d'un client, le serveur choisit aléatoirement une combinaison de 4 couleurs (avec éventuellement des répétitions).
- Le client lit les combinaisons entrées par l'utilisateur (sur l'entrée standard `System.in`) et les transmet au serveur.
- Après réception d'une combinaison, le serveur envoie au client deux entiers : le nombre de couleurs correctement placées et le nombre de couleurs qui sont dans la combinaison mais sont mal placées.
- Si le client a trouvé la bonne combinaison, le serveur affiche sur la sortie standard (`System.out`) la dernière combinaison envoyée, le nombre de tentatives effectuées par le client, puis ferme la connexion. Le client affiche aussi que la bonne combinaison a été trouvée avant de fermer la connexion
- Les échanges entre le client et le serveur se font en mode connecté : on utilisera donc TCP.

1.2 Code fourni

Une couleur est représentée par une lettre en capital. Les couleurs possibles sont 'B', 'G', 'O', 'R', 'W', 'Y' (pour *Black*, *Green*, *Orange*, *Red*, *White*, *Yellow*). Une combinaison est une chaîne formée avec les caractères 'B', 'G', 'O', 'R', 'W', 'Y'. Les répétitions sont possibles mais on se limitera à des chaînes de taille 4.

Les combinaisons sont représentées par des instances de la classe `Code` qui est fournie. Cette classe permet de créer une combinaison aléatoire ou à partir d'une chaîne de caractères. Les méthodes `numberOfColorsWithCorrectPosition(Code guess)` et `numberOfColorsWithIncorrectPosition(Code guess)` retournent respectivement le nombre de couleurs correctement placées et le nombre de couleur incorrectement placées dans la combinaison `guess`.

1.3 1. Client TCP simple

Écrire une première version du client qui :

1. établit une connexion TCP avec un serveur sur le port 1234
2. lit une chaîne de caractère au clavier
3. l'envoie ensuite au serveur.

1.3.1 Test fonctionnel

On pourra tester cette première version à l'aide de l'utilitaire `netcat`

```
$ netcat -l 1234
```

1.4 2. Serveur TCP simple

Écrire une première version du serveur qui :

1. attend des connections TCP sur le port 1234
2. une fois une connexion établie, génère une combinaison secrète aléatoire
3. affiche la chaîne reçue du client
4. affiche le nombre de couleurs correctement positionnées et le nombre de couleur incorrectement positionnées de la chaîne reçue par rapport à la combinaison secrète
5. envoie ces deux entiers séparés par un espace au client sous la forme d'une chaîne de caractères.

1.4.1 test fonctionnel

Tester le bon fonctionnement de votre serveur à l'aide de `netcat` et/ou en utilisant votre client simple.

1.5 3. Client et serveur complets

Compléter les codes client et serveur :

1. Ajouter une boucle dans le code client. A chaque itération de la boucle, lire une nouvelle combinaison au clavier, l'envoyer au serveur, recevoir le nombre de couleurs correctement et incorrectement positionnées et afficher ces deux nombres. La boucle se termine quand la bonne combinaison a été trouvée.
2. Modifier le serveur en conséquence (c'est-à-dire de façon à ce qu'il puisse interagir avec le nouveau client).
3. Faire en sorte que le serveur accepte de nouvelles connexions après que la bonne combinaison ait été trouvée par un client.

1.6 4. Améliorations

1. Traitement d'erreur. améliorer le code serveur pour gérer les erreurs : déconnexion inattendue du client, réception d'une combinaison non conforme, etc.
2. Que se passe-t-il si plusieurs clients essaient en même temps de se connecter au serveur ?