

M2-CCI Enseignement de Réseaux – Compte-Rendu du TP n.4

Programmation d'un jeu Mastermind avec gestion de la connexion

Groupe BAJA Sara – BOLLOLI Marco

Cahier de charges

Nous allons réaliser un jeu de Mastermind à travers la programmation de sockets d'un couple serveur/client en langage C.

Le principe est de deviner toutes les couleurs qu'a sélectionnés le maitre du jeu (le serveur) en moins de 10 tours. Nous avons réalisé l'implémentation pour un seul niveau de difficulté, avec 4 couleurs.

Les combinaisons de couleurs sont réalisées parmi les huit couleurs possibles :

1: bleu, 2: rouge, 3: vert, 4: jaune, 5: orange, 6: marron

Déroulement d'une partie :

Le client se connecte au serveur et le serveur lui répond avec un message qui explique le règles du jeu.

Le serveur choisi une combinaison de couleurs. Le joueur tente de deviner tour à tour la combinaison de couleurs.

A chaque tentative le maitre du jeu dévoile si chacune des couleurs est en bonne position, en mauvaise position, ou non présente dans la combinaison.

Le jeu de Mastermind se jouera en réseau connecté à un serveur. Plusieurs clients pourront jouer simultanément en étant connecté au même serveur. Le programme comprendra la gestion des erreurs de commandes entrées par le client.

Implémentation en réseau

L'application est placée coté serveur, le joueur interagit en émettant des requêtes client à destination du serveur.

Le serveur et le client communiquent grâce à des sockets TCP. Le serveur est à l'écoute en attente des requêtes du client.

Déroulement :

- client se connecte au serveur
- le serveur répond OK, il envoie les règles du jeu et demande
- demande la couleur de la case et envoie les codes couleurs
- le client rentre un chiffre correspondant à la couleur choisi
- si le chiffre est valide, le paquet est envoyé au serveur qui demande la couleur de la case suivante et envoie les codes couleurs
- si le chiffre n'est pas valide, le dernier message du serveur est réaffiché et le paquet avec la valeur erronée n'est pas envoyé.

- lorsque le client a donné une couleur à toutes les cases de la séquence, le serveur envoie un tableau récapitulatif avec 1 à l'emplacement des couleurs correctement placées, 0 à l'emplacement des couleurs présents mais mal placés ou un « vide » si la couleur n'est pas présente. Ce tableau récapitule toutes les couleurs rentrées par le joueur pour le tour en cours ;
- si toutes les couleurs sont correctement placées, le jeu est terminé, le serveur spécifie que la partie est gagnée au client puis ferme la socket TCP.
- sinon le nombre de tour est incrémenté et le serveur demande les couleurs d'une nouvelle séquence.
- Si à la fin du dixième tour le joueur n'a toujours pas trouvé la séquence correcte, la partie est perdue, le serveur le spécifie au client par un message et ferme la socket TCP.

Solutions choisis et protocoles utilisés

On utilise le protocole TCP/IP qui est un protocole orienté connexion et qui assure une connexion avant de lancer l'application. Ce protocole est le plus adapté au jeu qui ne nécessite aucune perte de message pour un déroulement optimale. Le protocole UDP n'a pas été retenu car l'application ne nécessite pas de réponses en « temps réel ».

On choisit d'utiliser un serveur parallèle car celui-ci est le plus adapté pour gérer les connexions de plusieurs clients simultanément. Avec ce type de serveur, un processus différent est dédié à chaque client. Ce choix est préféré au serveur itératif qui aurait généré de grands délais d'attente pour les différents clients. L'application est implémentée en langage C, ce qui facilite les appels système utiles à la duplication du processus de l'application.

Dans le déroulement du jeu, on décide d'indiquer les codes couleurs à chaque demande de couleur pour une case. Pendant un jeu, le serveur renvoie tous les séquences de couleurs essayés précédemment pour que le joueur puisse voir tous ses essais ainsi que la réponse du serveur concernant les emplacements des couleurs.