



Rapport Projet PSE

Réalisation du projet

Nous avons choisi le projet du jeu de coinche parce que c'est un jeu interactif pouvant très bien se jouer à distance. De plus, la réalisation de ce projet nécessite à la fois une partie programmation et une partie serveur.

Le projet s'est vite révélé être un projet très ambitieux, surtout en sachant que 2 membres du groupe ne connaissaient pas ce jeu.

Dans un premier temps, nous nous sommes concentrés sur la partie serveur dans un premier temps, car c'est cette partie qui se positionne le plus proche des cours de PSE. Les interactions entre le serveur et les clients sont donc possibles.

Par la suite, nous avons écrit l'intégralité des fonctions de la coinche sans se préoccuper du serveur. Ces fonctions écrivent et lisent tout sur un seul terminal. Il restait encore quelques points mineurs à implémenter comme les annonces (belotes/ rebelottes, le cent...) et d'autres fonctionnalités lorsque nous avons décidé d'implémenter l'usage du serveur dans le code du jeu.

Finalement, nous n'avons pas pu finir cette implémentation. Malgré tout nous avons fini les phases de préparation du jeu et la phase d'annonce des mises. des mises de chaque joueur.

Ces premières phases de la coinche illustrent tout de même bien le fonctionnement l'échange serveur/client que nous avons plus implanter préalablement.

Structure du Code

Le programme se découpe en 2 fichiers principaux et plusieurs headers.

Il y a tout d'abord le fichier contenant les différentes structures 'structure.h'.

Celui-ci contient les structures indispensable au fonctionnement de la partie (player, team, card, deck, bet).

Il y a ensuite le fichier 'serveur_coinche.c', contenant à la fois le code de la coinche et l'émission/réception des données des 4 clients (l'algorithme est fourni à la fin de ce rapport) avec son header.

Le fichier 'client_coinche.c' s'occupe lui de récupérer les infos permet aux 4 clients de se connecter indépendamment aux serveurs afin d'y émettre ou d'y recevoir des données avec son header.

Le fichier 'coinche.c' peut aussi être trouvé dans le gitlab. Il s'agit du code du jeu sans utiliser le serveur. Ce dernier vu une bonne première approche du jeu.

Fonctionnement de la partie serveur

Le serveur est fait de sorte que la communication entre le serveur et le client puisse être privé ou public.

Côté client, on gère la demande de pseudonyme à l'arrivée du client, puis lorsque tous les joueurs sont inscrits, la partie commence.

Côté serveur, on doit pouvoir communiquer avec tous les clients, mais aussi pouvoir cibler certains messages.

La distinction des clients est faite grâce aux `filedescriptors`, et chacune des informations relatives à chaque client est contenue la structure `player`, elle-même incluse dans un tableau `liste_joueur`.

L'envoi ciblé de messages du serveur au client se fait par la fonction `write` en y indiquant le `sockid` du client correspondant, cela est utilisé pour l'envoi des mains des joueurs, sans que les autres puissent le voir.

Algorithme de la coinche :

- Préparation de la partie

- Formation des équipes
- Créer une liste de l'ordre
- Choisir une tête de liste
- Affichage des règles
- Création d'un tas aléatoire (v,c) de 32 cartes

- Boucle tant que le score des 2 équipes est inférieur au score à atteindre :

- Distribution des 8 cartes :

- + Coupe du tas.
- * Génération un nombre aléatoire entre 5 et 27.
- * Coupe à cet endroit précis.
- + Distribution dans l'ordre du tas.

- Choix des mises (couleur + score)

- + chaque joueur propose (couple (couleur,mise (>80 et précédent)); coinche; passe) jusqu'à ce que 3 joueurs d'affilés ne proposent rien.

- Boucle, jeu en 8 tours (une carte par tour par joueur) :
 - Le premier tour le joueur de tête commence.
 - + Chaque joueur place une carte dans l'ordre de la table. On vérifié au fur et à mesure si la carte placée est meilleure que la meilleure précédemment placée. Si c'est le cas, elle devient la meilleure.
 - Si possible ajout de la restriction en fonction des règles (non implémenté).
 - Le joueur qui a placé la meilleure carte devient le premier joueur pour le tour suivant et on ajoute les cartes du tour au tas de son équipe.
- A la fin d'une donne, on calcule les points de chaque équipe et on vérifie si la mise est atteinte. On ajoute en fonction les points aux scores des équipes. Et on affiche les scores.
- Rotation dans l'ordre des joueurs pour qu'un nouveau joueur commence à miser.

Retour d'expérience

● Lucas Fichou

J'ai trouvé ce projet intéressant. Il m'a permis de progresser en C, un langage avec lequel je ne suis pas encore très à l'aise. J'ai aussi pu comprendre un peu mieux certains aspects du cours de PSE qui me faisait défaut notamment grâce à la pédagogie de mes camarades. La coinche est un jeu auquel j'aime bien jouer et essayer de la programmer fût une expérience mémorable.

● Marwan Fadel Abdallah

J'ai trouvé ce projet particulièrement intéressant d'une part sur la partie consistant à voir comment mettre le jeu de coinche sous forme de code et d'autre part appliquer les notions abordé vues en PSE.

Cependant j'ai rencontré des difficultés, m'étant rendue compte de mes lacunes en C et ma difficulté à appliquer les notions de PSE permettant la connection d'un utilisateur.

Malgré tout je compte utiliser cette expérience à bon escient afin de corriger ces lacunes.

● Nino Delien

J'ai trouvé ce projet PSE très intéressant parce qu'il intégrait à la fois le programmation en C et la partie réseau du cours de PSE. Ce fut un très bon complément de cours et ça m'a permis de me constituer de bonne base dans les 2 matières. J'ai bien aimé travailler sur le thème de la coinche. Je suis tout de même un peu déçu de ne pas l'avoir terminé dans les temps. J'ai tout de même l'intention de le rendre fonctionnel une fois que les autres projets pour les cours seront terminés car travaillé sur ce projet m'a beaucoup apporté.

● Zakaria El Gharmaoui

Le projet en soi m'a semblé intéressant et un cas concret d'application des méthodes vues en cours m'a permis de mieux comprendre la profondeur de ce qui a été appris, et que cela est la base de beaucoup de communications utilisés de nos jours. La distance a mis une vraie difficulté supplémentaire à ce projet, il n'était pas toujours facile de s'organiser pour la répartition des tâches, de trouver un horaire commun et on ne pouvait pas toujours programmer seul car il était nécessaire de bien comprendre tout ce qui avait été fait par les autres, ce qui nous bloquait souvent. Cela reste toutefois une bonne expérience, et savons que cela ressemble plus à ce que l'on retrouvera dans notre vie professionnelle qu'un simple exercice scolaire.