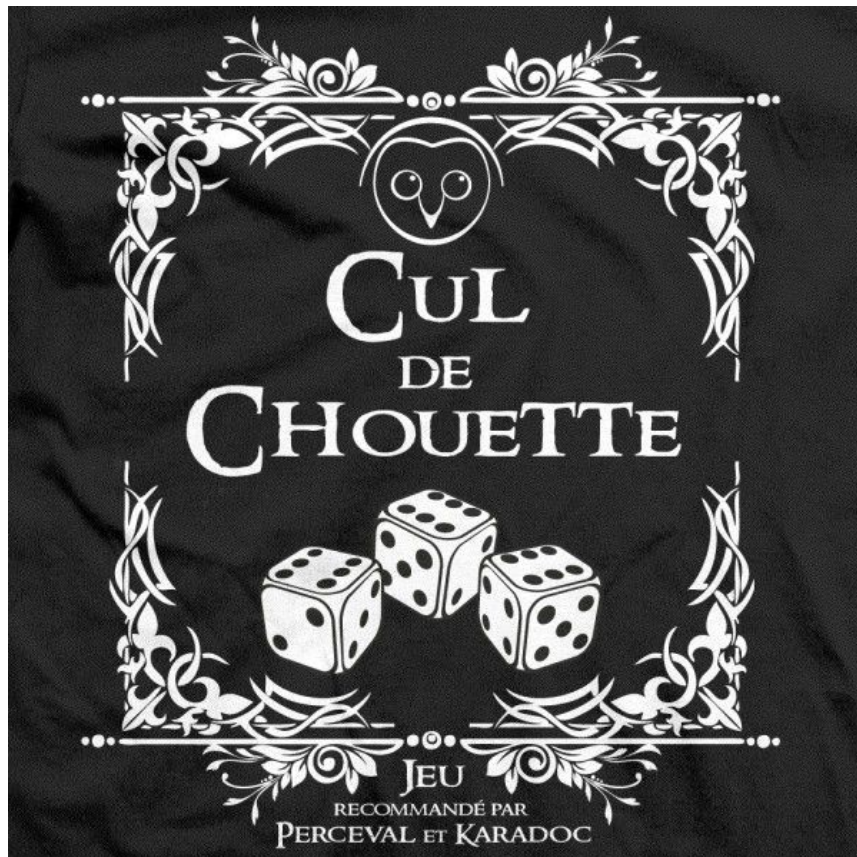


RAPPORT PROJET SYSTÈME DISTRIBUÉ



I/Introduction

Le but du projet est réaliser un programme gérant un jeu de dés, « le cul de chouette », inspiré de la série Kaamelott. C'est un jeu qui se joue à 3 joueurs, où chacun, à tour de lance lance deux dés puis le suivant pour marquer des points en fonction du score des dés, certain score requiert une interaction entre les joueurs afin de déterminer le gagnant. Le score à atteindre est normalement de 343.

Le Jeu est implanté avec un serveur en C multicast et des clients en C, un autre serveur en C enregistre les parties. L'objectif étant de mettre contribution les connaissances apprises en cours de manière concrète dans la réalisation d'un projet en binôme.

II/Réalisation

Le fonctionnement comporte 3 parties: la détection des joueurs, le déroulement de la partie, puis l'enregistrement des résultats

Afin de réaliser le projet nous avons commencé par déterminer les étapes pour la conception du projet, quelques règles de programmation, la répartition des tâches ainsi que les priorités.

Nous avons en premier réalisé le serveur multicast en C ainsi que les différentes fonctions qui lui sont propres afin qu'il puisse attribuer des identifiants uniques à chaque client. Les informations envoyées via le multicast sont définies par la structure suivante :

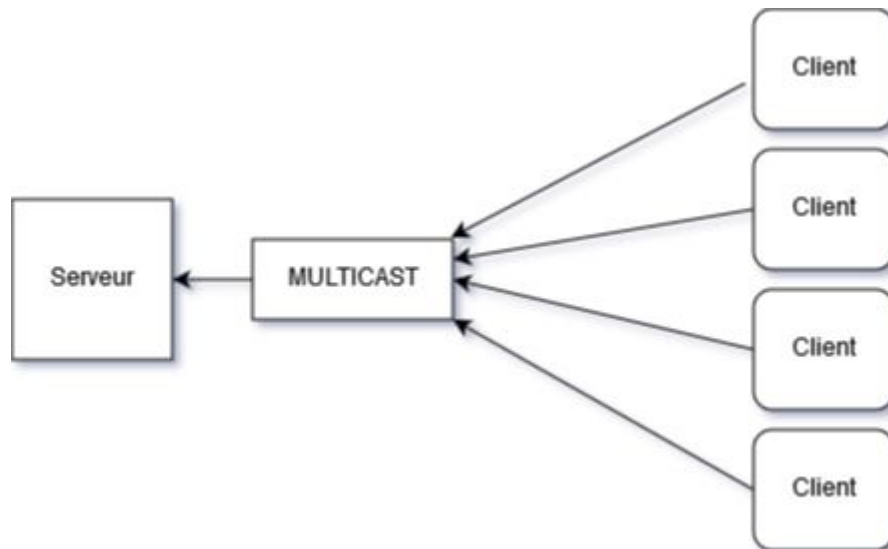
```
typedef struct
{
    char pseudo[BUF_PSEUDO];
    int id;
    struct sockaddr_in addr_client;
} multicast_request_t;
```

Nous avons ensuite créé les clients qui sont définis par une structure :

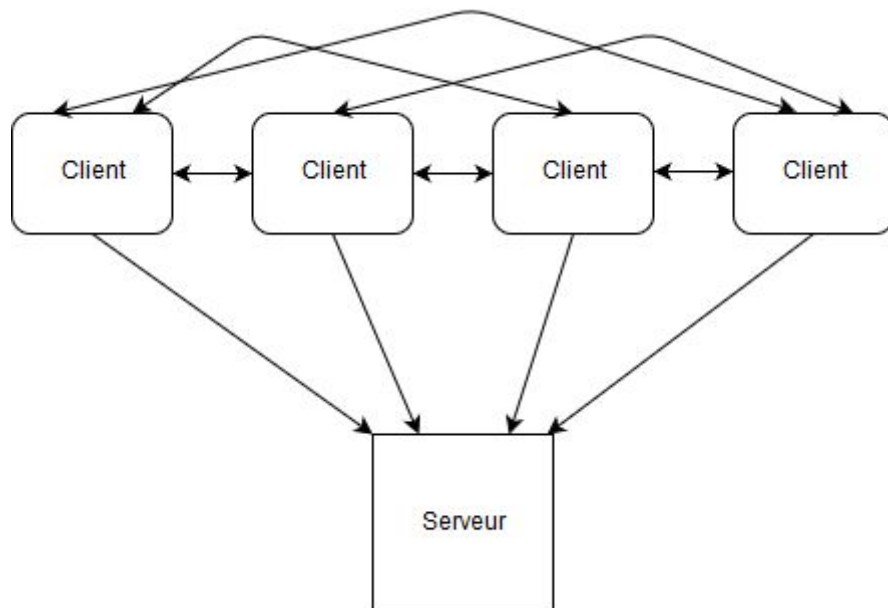
```
typedef struct
{
    int id;
    int score;
    int socket_client;

    des_t des[NB_DES];
    status_t status;
    char pseudo[BUF_PSEUDO];
} joueur_t;
```

Le serveur écoute sur le groupe multicast auquel chaque client est connecté.



Quand un nouveau client se connecte il récupère la liste des autres clients (via une connexion tcp avec le serveur) déjà connecté et établit une connexion tcp avec chacun d'entre eux, ainsi tous les clients auront la liste de tous les autres clients. Quand le client reçoit les informations sur les autres clients il se désabonne du serveur multicast.



Une fois que les quatre joueurs sont connectés entre eux, le premier joueur qui lance la partie devient l'hôte.

Quand nous avons réussi à gérer toutes les connexions entre les différents joueurs et le serveur, nous sommes passés au codage du jeu en tant que tel, en créant diverses fonctions pour contrôler les jets de dés, les différentes situations du jeu et le score.

III/ Conclusion

En définitive ce sujet a été une bonne mise en pratique des connaissances abordées lors des cours et TD/TP. Les plus gros challenges ont été en premier lieu le temps imparti très court et la gestion des connexion entre tous les clients.

Cependant l'intérêt que le sujet a suscité nous a permis de surmonter certaines de ces difficulté même si nous n'avons malheureusement pas pu finir faute de temps.