F) Fonctionnement global du projet

Cette partie va décrire le fonctionnement général du projet, c'est à dire de quelle manière l'ensemble des parties (des clients et du serveur) interagissent entre elles pour faire fonctionner le jeu.

Communication Client-Serveur:

Le but du projet est que le jeu de poker soit multi plate-forme. Le fonctionnement de chacun des clients doit donc être standardisé pour que le serveur fonctionne pour n'importe quel client et que si nécessaire, on puisse par la suite rajouter de nouveaux clients.

Pour que chaque client puisse communiquer avec le serveur, il faut que le type de message et les messages en eux-mêmes soient standardisés,

Les clients et le serveur communiquent donc avec des *Socket*, type d'objet disponible dans de nombreux langages. Chaque client et le serveur peuvent ainsi recevoir et comprendre les messages envoyés.

Pour que les messages de tous les clients puissent être traités de la même manière, nous utilisons un protocole strict (protocole en annexe), et les messages qui ne correspondent pas au protocole sont rejetés.

La communication se déroule en mode connecté, c'est-à-dire qu'un client doit se connecter au serveur, puis pourra communiquer avec lui.

De plus, le client ou le serveur peut fermer le circuit virtuel entre les deux pour diverses raisons comme perte de liaison internet, si un client est trop long à répondre présent lors d'un démarrage de partie etc...

Pour que le serveur puisse communiquer avec plusieurs clients, lorsqu'il reçoit la demande de connexion d'un nouveau client, il instancie une classe qui contiendra les informations du client connecté, et un *thread* pour pouvoir communiquer avec un client et continuer ses autres traitements.



Déroulement d'une partie:

Il y a trois types d'actions que peuvent effectuer les clients : les actions pré-partie ,les actions pour le commencement d'une partie, et les actions du déroulement de la partie.

Les actions pré-parties:

Chaque client peut, avant une partie, créer un compte, changer le mot de passe ou le pseudo du compte. Pour cela, le client se connecte, envoi les informations nécessaires puis est immédiatement déconnecté.

Pour qu'un compte soit créé, le pseudo et le mot de passe correspondant doivent respecter un format (de 2 à 15 caractères sans caractères spéciaux), et qu'un compte avec le même pseudo n'existe pas. La date de création du compte est conservée dans la base de données.



Les clients peuvent aussi se connecter physiquement au serveur, puis se connecter à leur compte en envoyant leur pseudo et leur mot de passe. Une fois « loggés » ils pourront effectuer les actions pour le commencement d'une partie.

Si un client se connecte alors qu'un client ayant le même compte est déjà loggé, alors ce dernier est déconnecté, permettant ainsi d’éviter les clients connectés et inactifs ainsi que d'avoir des clients au même pseudo.

Les actions pour le commencement:

Une fois « loggés », les clients peuvent :

-Demander des informations sur les parties en attentes (nom, nombre de joueurs).

-Demander des informations sur un autre client connecté.

-Créer une partie.

-Rejoindre une partie.

Une fois dans la partie, les clients reçoivent la liste des personnes présentes dans leur partie et attendent que le créateur de la partie la démarre.

Si le créateur de la partie quitte la partie, un autre client sera désigné comme créateur.

S’il n'y a plus personne dans la partie, elle est détruite.



Lorsque le créateur de la partie demande au serveur de lancer la partie (qui doit contenir au moins 2 clients), le serveur prévient tous les clients de la partie que la partie va commencer.

Les clients chargent leur interface de jeu puis envoient un message au serveur pour prévenir qu'ils sont prêts. Les clients qui mettent plus de cinq secondes à répondre sont expulsés de la partie.



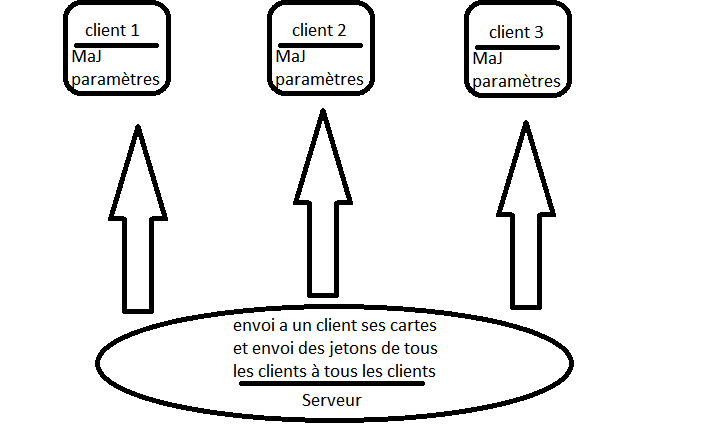
Les actions du déroulement de la partie :

La partie suit les règles du *Texas hold’em no limit* en mode tournoi. Pour résumer, le but du jeu est que tous les autres joueurs soient éliminés de la partie en perdant tous leurs jetons. Il y a 2 cartes distribuées à chaque joueur et 5 seront distribuées sur la table. La meilleure combinaison de 5 cartes gagne la manche. Il n’y a pas de limite de mise mais on ne peut relancer que 3 fois par tour d’enchère.

La partie commence par l’initialisation des jetons de chaque joueur à 200 et en mettant toutes les autres caractéristiques à 0 (pot, jetons posé sur la table, état (en jeu, tapis, perdu, couché)). Ensuite le serveur désigne qui va poser la *blind* et qui va poser la *surblind* (au début les 2 premiers de la liste) puis distribue les 2 cartes de la main de chaque joueur.

Chaque action concernant les joueurs fait l’objet d’un message envoyé aux clients pour qu’ils mettent à jour le leur de leur côté.

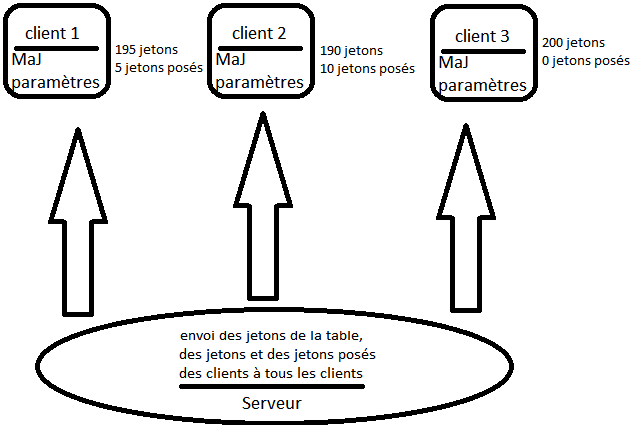
Le serveur envoi donc ici un message à chaque client avec ses cartes (celle que le client doit afficher dans la main du joueur) et jetons de chaque client à chaque clients, ici 200 pour que chaque client puisse afficher les jetons de tous les joueurs.



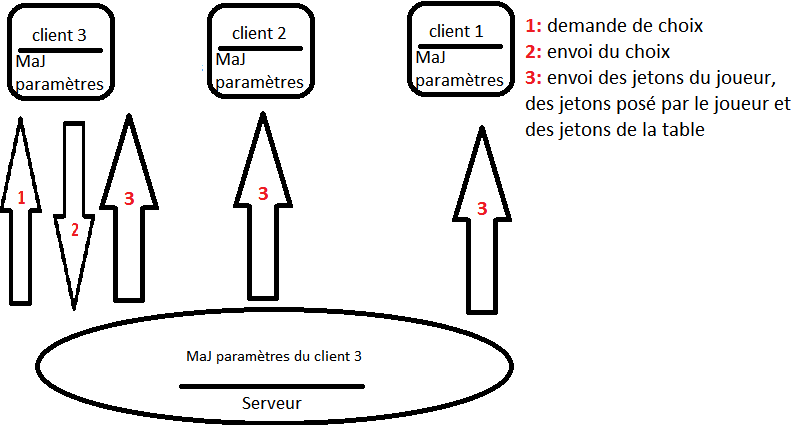
Nous pouvons maintenant commencer le premier tour de jeu (on va définir le tour de jeu comme commençant après que les deux cartes de la main soient distribuées et se terminant après la comparaison des mains (s’il y en a besoin) et la répartition des jetons au(x) gagnant(s) du tour).

La blind et la surblind sont posées par les joueurs désignés (5 et 10 jetons). Le serveur retire donc 5 et 10 jetons à ces joueurs et ajoute ces mêmes valeurs au nombre de jetons qui sont sur la table. On a donc maintenant 15 jetons sur la table et les deux premiers joueurs ont 195 et 190 jetons et 5 et 10 jetons posés.

Toutes ces informations sont donc envoyées à chaque client pour leur permettre de mettre à jour leurs données.



Ceci s’enchaine avec le premier tour d’enchère. On fait le tour de chaque joueur en commençant par le joueur qui suit la surblind (ici le client 3). On lui demande ce qu’il veut faire en lui envoyant un message avec ses choix possible et les jetons minimum et maximum qu’il peut miser. Le client a le choix entre se coucher, suivre/check ou relancer. Le serveur se met en attente du choix du joueur pendant 15 secondes. Si au bout de cette attente le client n’a pas renvoyé de message de choix, le serveur le fait se coucher sinon il applique le choix du joueur. Ensuite le serveur envoie un message pour indiquer les modifications des paramètres du joueur et on passe au joueur suivant et on lui demande son choix et ainsi de suite.



A la fin du tour d’enchère, 3 cartes sont tirées pour la table et envoyées aux clients et on refait un tour d’enchère. Puis on continue en tirant une carte, un tour d’enchère, la dernière carte et le dernier tour d’enchère. Bien sûr, les tours d’enchères ne se font que si nécessaire (encore au moins 2 joueurs peuvent miser (c’est-à-dire qu’ils ne sont ni couché, ni en tapis, ni défait)).

S’il ne reste qu’un seul joueur qui n’est pas couché, alors il gagne les jetons de la table. Sinon, il faut voir qui est le gagnant du tour de jeu. Le serveur alors calcule qui est le gagnant ou qui sont les gagnants en cas d’égalité puis met à jour les données des clients et envoie un message à un client pour leur indiquer quelles sont les cartes des autres joueurs, qui est le gagnant et les changements effectué sur les jetons de chacun. En cas de tapis, si le client gagne, il ne recevra que ce qu’il a le droit de toucher (jetons misés par les clients inférieur ou égal à son tapis) et le reste des jetons est réparti sur les autres gagnants ou les joueurs qui ne sont pas couchés.

A la fin de chaque tour de jeu, on vérifie si des clients se retrouvent avec zéro jeton. Si c’est le cas alors ces joueurs ont perdu et ne peuvent plus jouer, mais ils peuvent rester en tant que spectateur. A chaque fois qu’il y a un nouveau perdant le serveur envoie un message à tous les clients pour leur indiquer.

S’il ne reste plus qu’un seul joueur en jeu alors la partie se termine et ce joueur gagne la partie. On envoie donc un message à tous les clients pour leur indiquer le gagnant et la partie est détruite.

Par contre, s’il reste plusieurs joueurs en jeu, un nouveau tour de jeu se lance et on recommence en assignant la mise de la blind au joueur suivant l’ancienne blind.