

Protocol Server Rtype

Notre serveur permet à “n” joueurs sur le même réseau de pouvoir se connecter et de se retrouver à l’aide de parties. Les joueurs connectés peuvent décider de créer une partie en lui mettant un nom ainsi que le nombre de joueur qu’ils souhaitent avoir dans leur partie. Sinon ils peuvent voir les parties en cours ainsi que celle disponible et peuvent les rejoindre. Une fois qu’une partie est complète, celle-ci se lance et les clients pourront se déplacer en utilisant les flèches.

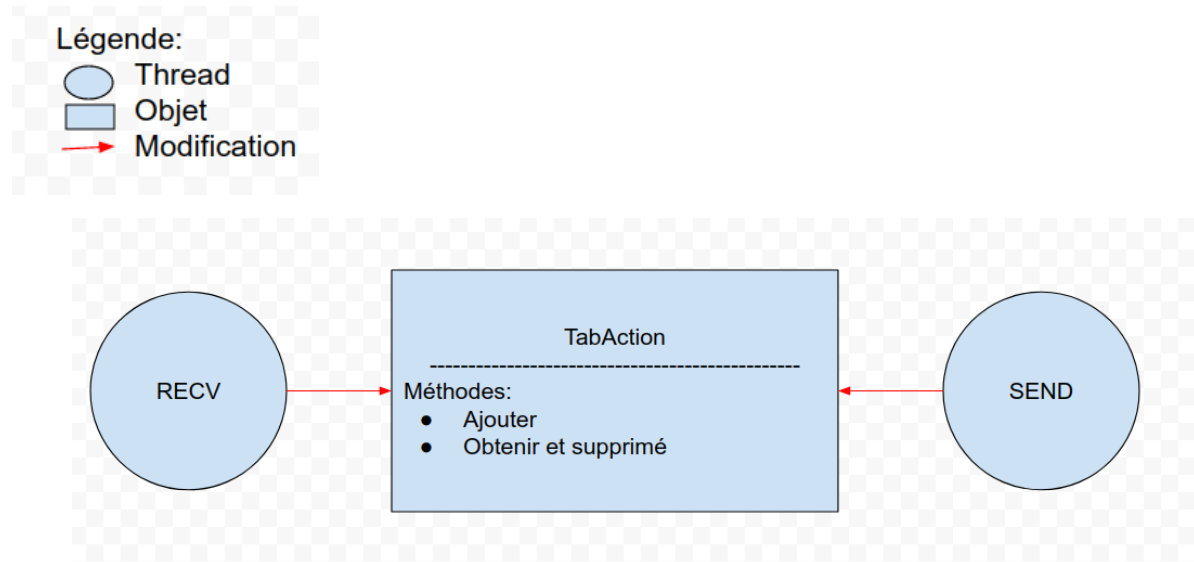
Notre server ouvre un socket sur un port et permet de recevoir une commande de n’importe quelle adresse étant sur le même réseau. Les commandes acceptées sont :

- “connect [pseudo]” -> se connecter avec un pseudo
- “create [name_party] [nb_player_party]” -> créer une partie avec un nom ainsi que le nombre de joueur. Le joueur rejoint aussi la partie.
- “join [name_party]” -> rejoindre une partie
- “display” -> affiche les parties disponibles et en cours
- “leave” -> quitter la partie (avant que celle-ci soit lancée)
- “quit” -> permet de supprimer votre pseudo du server (quitter le serveur)

Nous utilisons le “main” pour la gestion des commandes serveur. Après chaque réception de commande, nous renvoyons un code au client, lui permettant de savoir si sa commande a été acceptée. Lorsqu’un joueur rejoint une partie ou en créer une, il reçoit les ports sur lesquels communiquer afin de pouvoir jouer. Dès lors qu’une partie est complète, le server lance deux threads (un pour recevoir les déplacements et un autre pour faire tourner la partie ainsi que renvoyer toutes les informations concernant le jeu aux clients). Nous utilisons donc deux threads par partie lancée.

Prenons l’exemple d’une partie lancé possédant “n” joueurs :

Le thread de réception reçoit les commandes des “n” clients, les transforment en “Action” et les ajoute dans une liste “TabAction”. Le 2e est le thread envoie et gère la partie. Il possède aussi la “TabAction”, il récupère les différentes actions, les exécute, puis il fait tourner le jeu, et renvoie à tous joueurs les informations à afficher côté client. Puisque les deux threads modifient une même classe, les méthodes de modification de celle-ci possède un mutex qui permet de gérer les problèmes de concurrence. Lorsqu’il n’y a plus aucun joueur de connecter sur cette partie, les threads se ferment.



Lorsqu'un joueur quitte une partie déjà lancé, son avatar disparaît de celle-ci et il ne pourra plus la rejoindre de nouveau. Lorsqu'une partie ne compte plus aucun joueur, celle-ci est détruite.

Voici les codes que les clients recevront :

Connexion :

- 100 connexion acceptée
- 300 quitter accepter
- 401 impossible de se connecter sans pseudo

Parties :

- 201 création de partie accepté
- 402 création de partie refusé (nom ou nombre personne manquante)
- 202 vous avez rejoint une partie
- 403 impossible de rejoindre une partie sans mettre de nom
- 204 vous avez reçu les différentes parties disponibles et en cours
- 203 vous avez quitté une partie si vous étiez dans une.

Rtype

Légende :

