

ROB4

Compte-rendu

Jeu etu13

Léo Laval – Lucas Ribeiro
20/01/2022

Explication du jeu

But du jeu :

Démasquer le meurtrier parmi les 13 suspects.

Contenu du jeu :

13 cartes uniques.

Chaque carte est composée d'un nom et de 2 à 3 symboles.

Chaque combinaison est unique.

Déroulement de la partie :

Chaque joueur reçoit 3 cartes.

A tour de rôle, les joueurs ont 3 choix:

- Demander à tous les joueurs, s'ils possèdent au moins un symbole donné parmi les trois cartes en leurs possessions. Dans ce cas, la réponse de chaque joueur est soit oui, soit non. (Le nombre de symboles demandés possédés n'est pas spécifié.
- Demander à un joueur unique le nombre d'un symbole donné qu'il possède.

(Dans ces deux cas, la réponse est globale. Chaque joueur obtient les mêmes informations.)

- Accuser le meurtrier. Si le joueur a raison, il a gagné.

Ce qui a été implémenté pour que le jeu fonctionne :

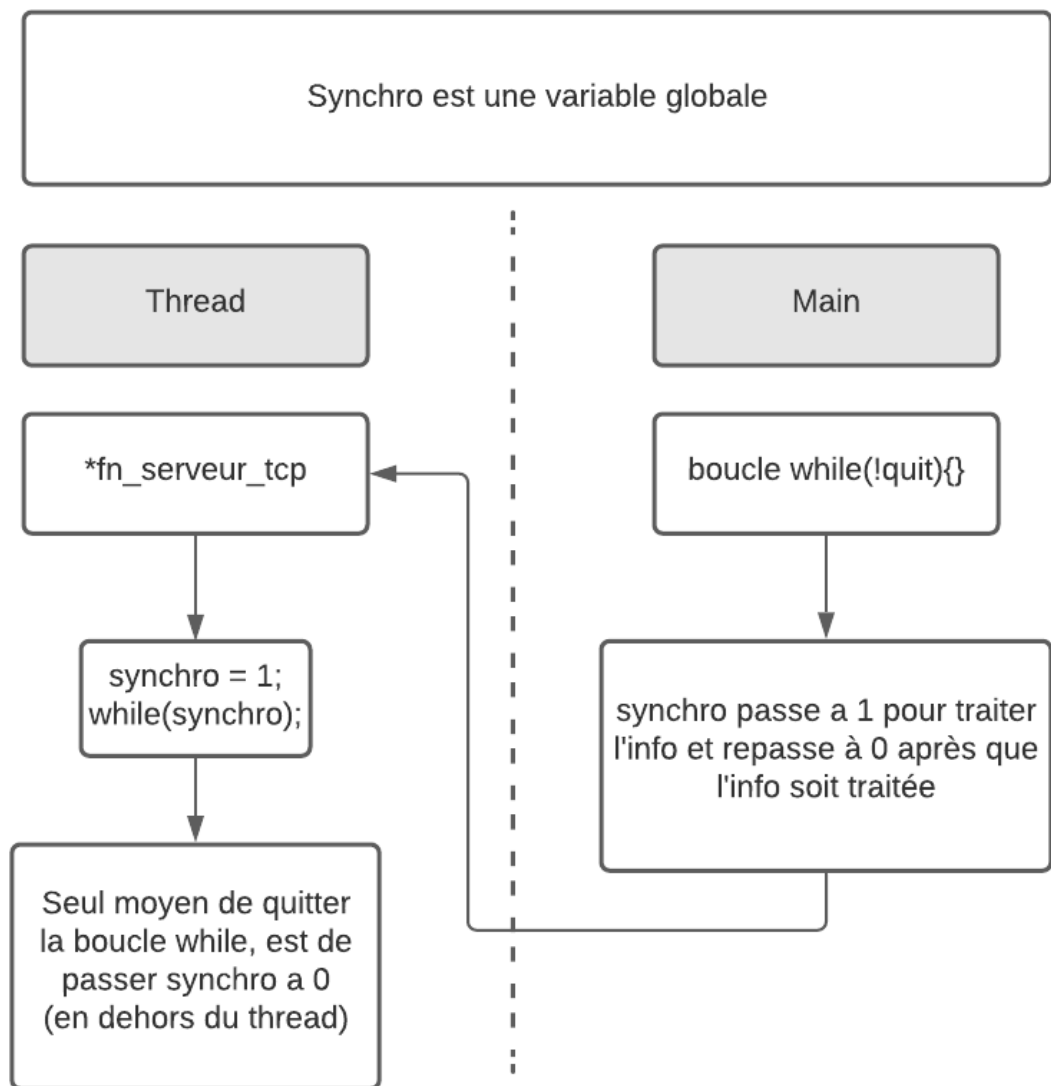
- Réception du joueur ID
- Réception de la liste des joueurs
- Réception des cartes
- Réception de l'ID du joueur courant

Activation du bouton Go chez un joueur s'il possède le même ID que l' ID courant. Le bouton n'est cliquable que si ce que l'utilisateur à sélectionner est valide.

- Réception des réponses aux questions posées:
 - Nombre exact d'un symbole pour un joueur
 - Possession ou non d'un symbole chez les joueurs

Remarque : Il faut faire attention à ne pas remplacer une valeur connue au préalable par une étoile. L'étoile signifie la possession du symbole chez un joueur.

Explication de synchro



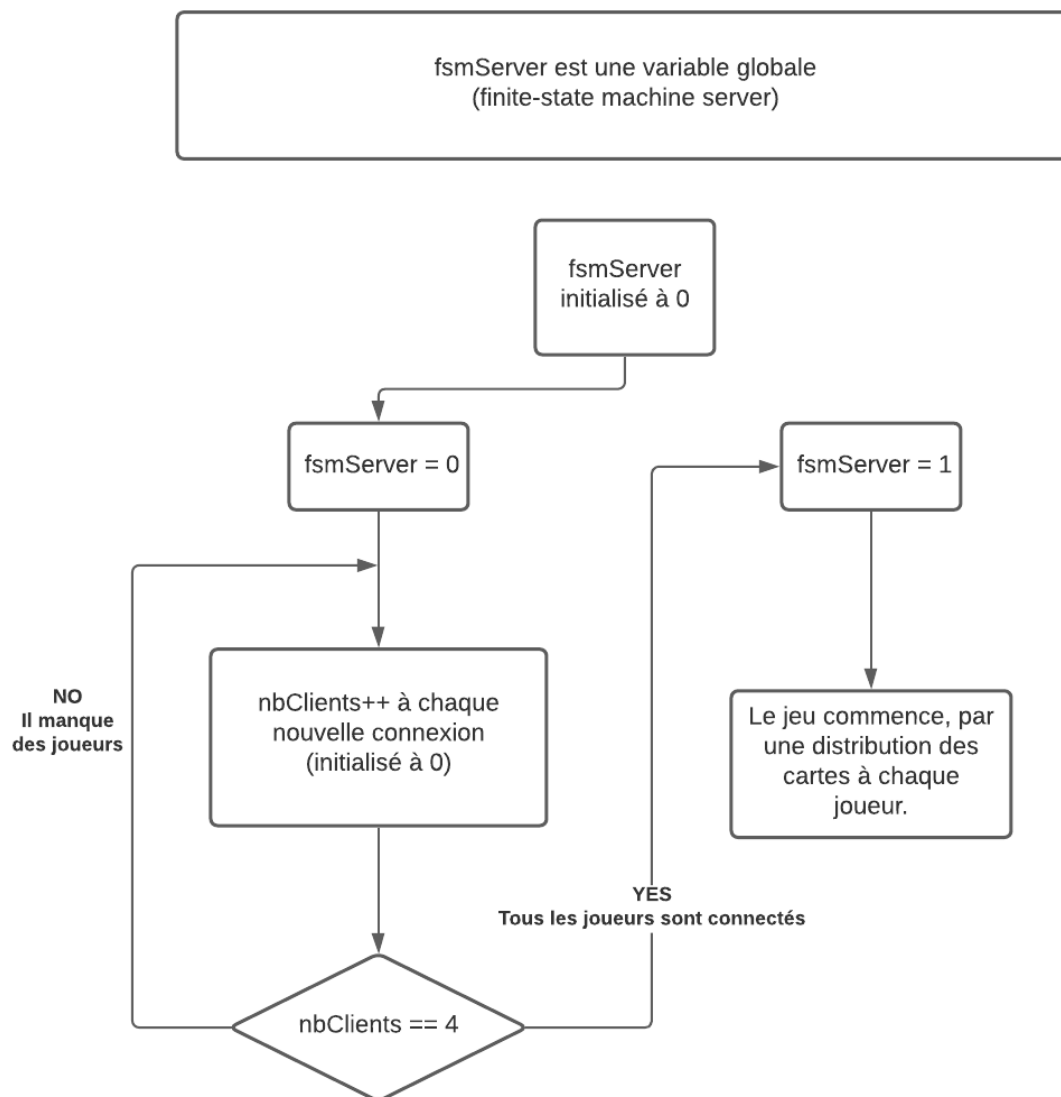
La variable synchro passe à 1 dans le serveur thread du client quand il reçoit un message du serveur. Ce message est ensuite traité par le client dans le main. Une fois que ce message a été traité, synchro repasse à 0 dans l'attente d'un nouveau message.

Cette variable permet 2 choses :

- Le client ne traite le buffer qu'une seule fois
- Le serveur du client attend que le message soit traité pour en recevoir un nouveau

Remarque: l'utilisation du thread est nécessaire car l'attente de réception d'un message est une opération bloquante.

Explication de la variable fsmServer



La variable `fsmServer` permet de détecter s'il y a un nombre suffisant de joueurs pour démarrer une partie. Tant qu'il n'y a pas 4 joueurs, le serveur attend de nouveaux joueurs. Une fois cette condition remplie, il envoie les cartes à chacun des joueurs, et la partie peut commencer.

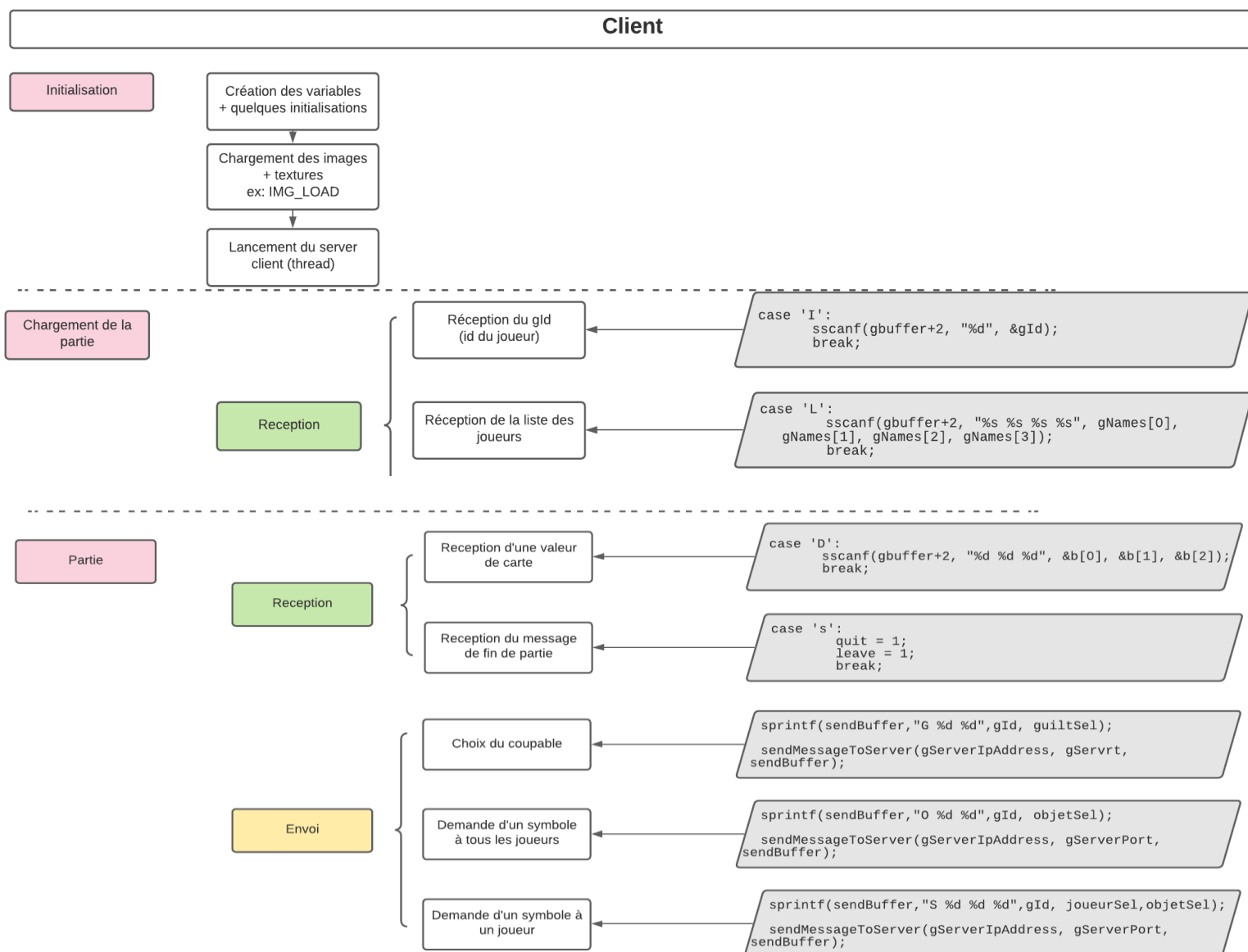
Ajouts par rapport au jeu fourni par le professeur:

- Le serveur et les clients se ferment lorsqu'il y a un vainqueur. Un message général est envoyé à tous les clients.
- Lorsque le client appuie sur le bouton go, les colonnes et les lignes qu'il a sélectionnées se désélectionnent.
- Le jeu est désormais aléatoire, le tirage n'est pas connu à l'avance (sauf si un filou regarde le terminal du serveur).
- Option : ajout d'une image de victoire/défaite pendant un laps de temps de 10s avant la fermeture du jeu.

Diagramme UML:

• Client

:



• Serveur :

