

Projet final - Client/serveur

Connect 4

Patrick Doucet A00177291

Gabrielle Gionet A00173009

Jonathan Godin A00167329

Présenté à : Mustapha Kardouchi

Dans le cadre du cours: INFO4002

Pavillon Rémi-Rossignol

Université de Moncton

10 Décembre 2018

Introduction

Nous devons développer une application client-serveur qui utilise les threads et les sémaphores. Nous avons choisi de faire un jeu de Connect 4. Le jeu consiste en une grille de 7x6 cases avec des jetons rouges et noirs. Les joueurs placent un jeton chacun leur tour et le gagnant est celui qui obtient une séquence de 4 jetons dans une ligne verticale, horizontale ou diagonale avec sa couleur. Pour notre application, le côté client est codé en C#, et le côté serveur est codé en C++.

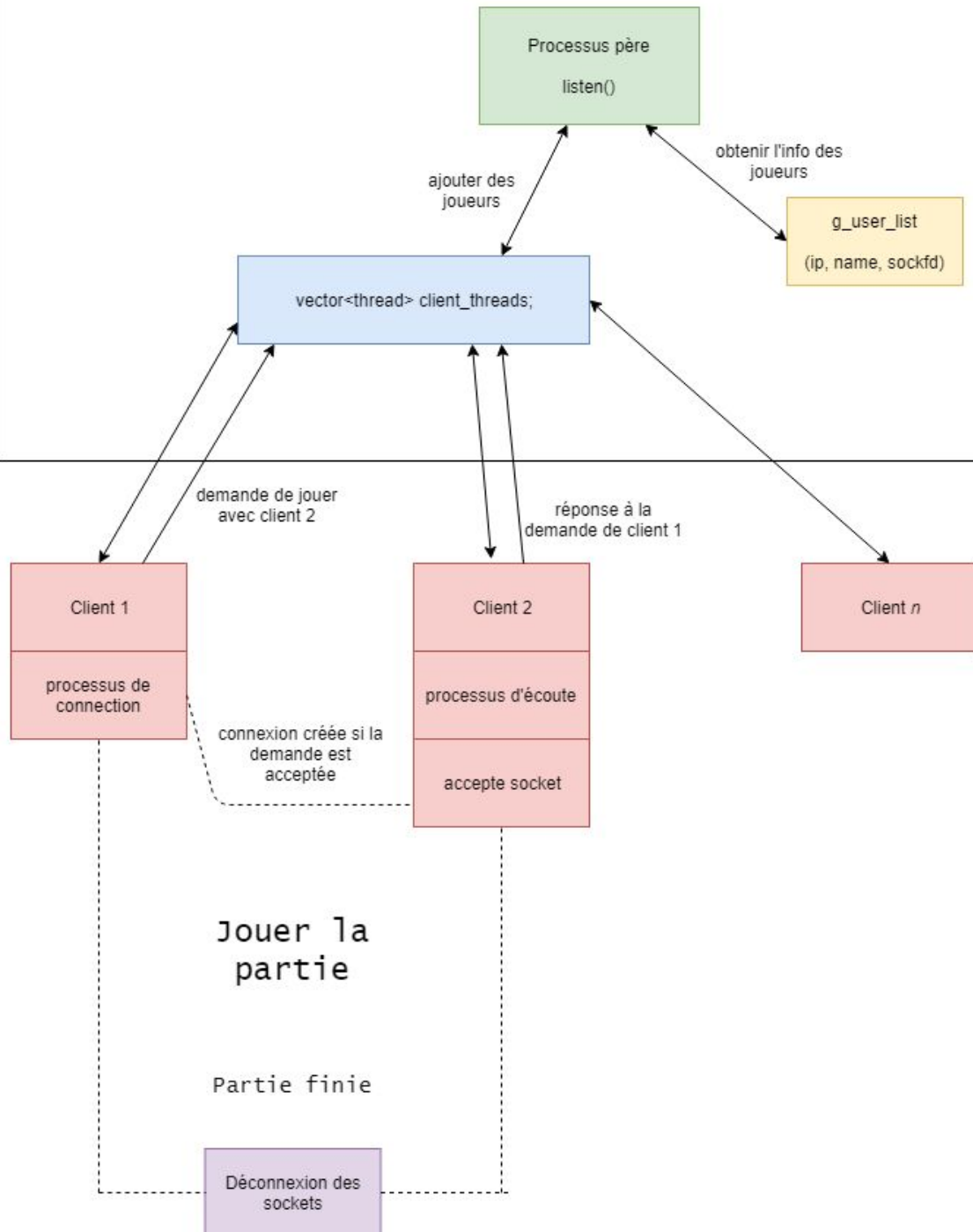
Description et schéma

Le programme serveur est en mode parallèle. Le serveur écoute pour des nouvelles connexions. Quand un client se connecte, il est ajouté à une liste globale qui contient l'information (adresse IP, nom et identificateur du socket) de tous les joueurs connectés. Un thread est créé pour écouter sur le socket du client pour la prochaine information reçue du client, qui sera soit un nom d'un client contre qui il veut jouer, soit un message de déconnexion ou la réponse à une demande de jeu provenant d'un autre client. Dans le cas où un client veut jouer contre un autre client, un des clients envoient au serveur le nom d'un client contre qui il veut jouer. Le serveur envoie une demande à ce dernier. Ce client va envoyer une réponse, soit accepter ou refuser. Le serveur transmet la réponse à celui qui demandait. Si la partie est lancée, les clients avisent le serveur qu'ils vont se déconnecter. Le serveur peut donc les enlever de sa liste de clients, ce qui les fera disparaître de la liste de clients des autres clients connectés.

Lorsque le jeu est lancé, le client qui a répondu s'il voulait jouer contre l'autre crée un processus d'écoute pour son adversaire. L'adversaire connecte au client en utilisant l'IP du client. Le client accepte alors la connexion et les deux joueurs commencent le jeu.

La seule information qui est envoyée entre les deux clients durant la séquence du jeu est la colonne où ils ont appuyé. Il traite indépendamment cette information pour déterminer l'état de leur grille et s'il y a eu un gagnant. Si la partie est terminée, les joueurs sont donnés l'option de jouer une autre manche, ou de terminer la session. Si la session est terminée, les clients se déconnectent de leur socket, et l'application est fermée.

Serveur



Notes pour la compilation:

- Compiler le serveur “.\Server-Side\Connect-4-Server.cpp” **sur linux** en utilisant g++ -pthread
- Si vous avez les packages c# sur visual studio 2015/2017, vous pouvez compiler la solution du client, ou exécuter le .exe dans “.\Client-Side\Connect-4-Client\Connect-4-Client\bin\Debug\Connect-4-Client.exe”.
- **Le client doit être compilé/exécuté sous windows.**

Notes pour les fonctions importantes du côté client:

Pour ne pas avoir besoin de chercher pour les fonctions qui traitent les sockets ou qui font des interactions avec ceux-ci.

Les fonctions importantes qui font des communications avec le serveur sont:

- [MainForm.cs]
 - ◆ StartClientSocket()
 - ◆ sendNameToServer()
 - ◆ fetchClientListFromServer()
 - ◆ AlwaysListening()
 - ◆ IsSocketConnected()
 - ◆ CloseClientSocket()
- [AcceptForm.cs]
 - ◆ button1_Click()
 - ◆ button2_Click()

Les fonctions importantes qui font des communications avec d'autres clients sont:

- [GameForm.cs]
 - ◆ GameForm()
 - ◆ GameForm_MouseClick()
 - ◆ isReset()
 - ◆ closeGame()