Peer To Peer :

Using :



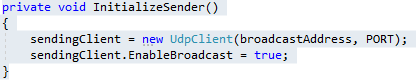
Il faut definir un délégué « delegete void addMessage(string message) ; » ca va servir a encapsuler les messages en mode FIFO.

Il faut un receivingClient et un sendingClient.

Les 2 sont des « UdpClient ».

Un UdpClient peux interagir avec le réseau via une adresse et un port.

Le sendingClient va être l’outil qui va nous permettre d’envoyer les messages.

Il faut l’initialiser donc lui fournir les informations nécessaires comme l’adresse et le port

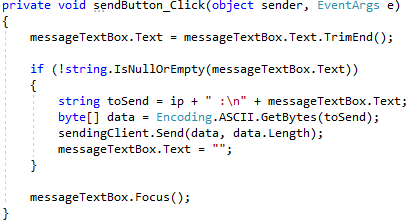
broadcastAddress est l’adresse de diffusion « 255.255.255.255 » va très bien.

PORT est le port utilisé donc un port libre la liste se trouve sur internet.

La méthode ci-dessus doit être lancé au lancement de l’application.

Send Button :

La méthode ci-dessous est la méthode d’envoi d’un message.

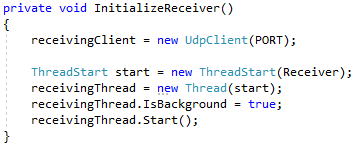


La chose à retenir est qu’il faut encoder le message via la table Ascii avant de l’envoyer.

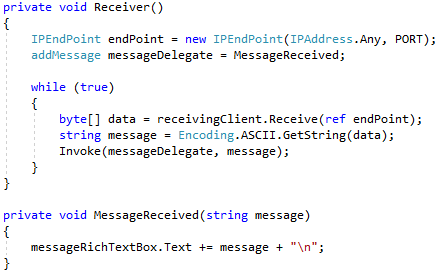
On envoie le message et on espère qu’il y a un autre réceptionneur sur la même adresse et port.

receivingClient :

Le receivingClient doit être initialisé comme le sendingClient donc il doit aussi etre initialisé au lancement de l’application.



On doit lui donner le numéro du port (le même que le sendingClient). On va créer un thread pour que le receivingClient soit en permanence en fonction sans bloqué l’application. La méthode Receiver est ci-dessous.



L’IPEndPoint est défini le point de fin du réseau

addMessage est un liste qui garde les messages dans la mémoire comme une file FIFO.

Un boucle infinie est créé est récupère les messages envoyé depuis un autre poste puis l’ajoute au texte d’une richTextBox.

Webographie :

http://www.dreamincode.net/forums/topic/231058-peer-to-peer-chat-advanced/