UNIVERSITÉ D'AVIGNON
ET DES PAYS DE VAUCLUSE

C E N T R E
D'ENSEIGNEMENT
ET DE RECHERCHE
EN INFORMATIQUE

Licence Informatique
ngénierie du Logicie
UE projet de programmation

>>> # Rapport01

soufiane ESSABONI, Ibtissam BENACHOUR

13 octobre 2015

CERI - LIA
339 chemin des Meinajariès
BP 1228
84911 AVIGNON Cedex 9
France

Tél. +33 (0)4 90 84 35 00
Fax +33 (0)4 90 84 35 01

http://ceri.univ-avignon.fr

Table des matières

Titre 1

Table des matières 2

1 présentation de la composante réseaux 3

2 interaction avec les autres composantes 3

3 les classes 3
- 3.1 classe InetAddress : . . . 3
- 3.2 La classe Socket : . . . 3
- 3.3 La classe ServerSocket : . . . 3
- 3.4 La classe MulticastSocket : . . . 3
- 3.5 La classe DatagramSocket : . . . 4

4 les méthodes 4
- 4.1 les méthodes de la classe InetAdress . . . 4
- 4.2 Les méthodes de la classe Socket . . . 4
- 4.3 les méthodes de la classe ServerSocket . . . 4
- 4.4 les méthodes de la classe DatagramSocket . . . 4

Références 5

1 présentation de la composante réseaux

Java est un langage de programmation orienté objet développé par Sun. En plus d'un langage de programmation, Java fournit également une très riche bibliothèque de classes pour tous les domaines d'application de l'informatique,et d'une part on va aborder le sujet de la composante de notre projet qui est le moteur réseaux qui represente un mécanisme permettant l'appel de méthodes entre objets JAVA s'exécutant sur des machines virtuelle différentes. Le rôle de ce moteur se traduit par la relation entre l'interface utilisateur et le Joueur Network qui se fait par la transmission des nécessaires à afficher.et l'inverse par le joueur qui transmets au moteur les ordres de l'utilisateurs.

- Comment ça se passe ?

Le Joueur network attend la connexion de l'interface utilisateur avec un ServerSocket, et la communication peut se faire soit via un flux de caractères(xml,txt...) soit en serialisant des objets ,si non les deux a la fois..et l'avantage de l'utilisation de moteur concerne la possibilité d'un joueur a utiliser le jeu à distance via le réseau elle inclut donc la création d'une partie ouverte aux connexions distantes.

2 interaction avec les autres composantes

L'implémentation du joueur est fournie par l'interface utilisateur qui represente l'ensemble des menus qui permettant a l'utilisateur de configurer et lancer une partie, et qui caractérise les détailles du jeu par chaque joueur et les transmettre au moteur physique ,ce dernier traite les etats des objects,et la detection des collisions dans le but d'être capable de recevoir toutes les demandes des joueur et de travailler sur ces etats par une façon appropriée, afin d'avoir le résultat sur l'ecran après la récuppération des données par le moteur graphique qui interagisse avec l'utilisateur et assurele dialogue entre eux.cette interface permet d'utiliser des différents modes d'affichage pour avoir une propre vue sur le jeu,Le choix de cette composante était sur la base de travailler le langage Java et acquérir en même temps des nouvelles connaissance sur le fonctionnement du réseaux en relation avec le Java.

3 les classes

3.1 classe InetAddress :

La classe java.net.InetAddress permet de représenter les adresses IP. Chaque objet de cette classe possède deux champs hostName et address contenant respectivement une chaîne de caractère et un tableau d'octets.

Le champ hostName stocke le plus souvent le nom de l'hote et le champ address l'adresse IP

3.2 La classe Socket :

La classe Socket représente en Java les sockets utilisés côtés client ou les sockets de service.

3.3 La classe ServerSocket :

Cette classe permet de créer des sockets qui attendent des connections sur un port spécifié et lors d'une connexion retournent un Socket qui permet de communiquer avec l'appelant.

3.4 La classe MulticastSocket :

Cette classe permet d'utiliser le multicasting IP pour envoyer des datagrammes UDP à un ensemble de machines repéré grâce à une addresse multicast (classe D dans IP version 4 : de 224.0.0.1 à 239.255.255.255).

3.5 La classe DatagramSocket :

Cette classe crée un Socket qui utilise le protocole UDP (Unreliable Datagram Protocol) pour émettre ou recevoir des données.

4 les méthodes

4.1 les méthodes de la classe InetAdress

cette classe propose les méthodes suivantes :

- InetAddress GetByName(Type string) => Renvoie l'adresse internet associée au nom d'hôte fourni en paramétre.
- InetAddress[] getAllByName(String) => Renvoie un tableau des adresses internet associées au nom d'hôte fourni en paramètre.
- InetAddress getLocalHost() => Renvoie l'adresse internet de la machine locale.
- byte[] getAddress() => Renvoie un tableau contenant les 4 octets de l'adresse internet.
- String getHostAddress() => Renvoie l'adresse internet sous la forme d'une chaîne de caractères .
- String getHostName() => Renvoie le nom du serveur.

4.2 Les méthodes de la classe Socket

cette classe propose les méthodes suivantes : - InetAddress getInetAddress() => Renvoie l'adresse I.P. à laquelle la socket est connectée

- void close() => Renvoie l'adresse I.P. à laquelle la socket est connectée
- InputStream getInputStream() => Renvoie un flux en entrée pour recevoir les données de la socket
- OutputStream getOutputStream() => Renvoie un flux en entrée pour recevoir les données de la socket
- int getPort() => Renvoie un flux en entrée pour recevoir les données de la socket

4.3 les méthodes de la classe ServerSocket

cette classe propose les méthodes suivantes : - Socket accept() => Attendre une nouvelle connexion

- Socket accept() => Fermer la socket

4.4 les méthodes de la classe DatagramSocket

La classe DatagramSocket propose aussi plusieurs Méthodes : - close() => Fermer la socket et ainsi libérer le port

- receive(DatagramPacket) => Recevoir des données
- send(DatagramPacket) => Envoyer des données
- int getPort() => Renvoyer le port associé à la socket.
- void setSoTimeout(int) => Préciser un timeout d'attente pour la réception d'un message.

Références

http ://torguet.net/cours/Sockets/SupportSockets.pdf