

# GameServer : Mode d'emploi

## Présentation de la classe

GameServer gère la communication avec le serveur. En cas de besoin, elle peut également démarrer un serveur local pour vos tests, auxquels d'autres pourront se connecter.

Elle se situe dans le package gameserver

GameServer
<pre>static GameServer startLocalGame() static GameServer startRemoteGame     (String host, int port) login(String name) throws IOException logout() sendCommand(String c) throws IOException String readResult() throws IOException</pre>

startLocalGame() démarre un serveur local sur le port 12123 et s'y connecte

startRemoteGame(h,p) se connecte à un serveur existant à l'adresse h, sur le port p

login(s) s'identifie sur le serveur

logout() se déconnecte du serveur

sendCommand(c) permet d'envoyer une commande ("map", "move S", "scout", etc...)

readResult() lit une ligne en provenance du serveur. Attention cette fonction est bloquante tant qu'il n'y a rien à lire

## Commandes

Les commandes sont décrites dans le cahier des charges du jeu à une exception près : list  
En effet, il faut détecter le dernier nom de la liste. Pour signaler la fin, le serveur envoie un "OK".

Pour le reste des commandes, tout fonctionne comme décrit.

## Exemple d'utilisation

```
GameServer gs;

try {
    gs = GameServer.startLocalGame();

    int x,y;

    gs.login("test");

    gs.sendCommand("position");
    String s = gs.readResult();
    String[] pos;

    s = s.substring(1,s.length()-1);
    pos = s.split(";");
    x = Integer.parseInt(pos[0]);
    y = Integer.parseInt(pos[1]);

    gs.sendCommand("loadout");
    s = gs.readResult();
    int load= Integer.parseInt(s);

    gs.sendCommand("list");

    String list = "";
    s = gs.readResult();
    while(!s.equals("OK")) {
        list = list + s + '\n';
        s = gs.readResult();
    }

    gs.sendCommand("move S");
    String moveRes = gs.readResult();

    gs.logout();
} catch (GameAlreadyStartedException e) {
    e.printStackTrace();
} catch (IOException e) {
    e.printStackTrace();
}
```

## **Note importante**

malgré tous mes efforts, il se peut que quelques bugs subsistent, le programme étant assez complexe. Surveillez donc les mises à jour éventuelles.

Par ailleurs, vous verrez des classes dans la librairie qui peuvent ressembler à ce que vous voulez faire ou utiliser. Evitez d'essayer d'utiliser autre chose que la classe GameServer, les autres sont conçues pour faire fonctionner le serveur, et pas d'éventuels clients.

## **Intégration dans votre projet**

Pour ajouter cette librairie dans un projet Eclipse, allez dans les propriétés du projet, dans la zone "Build Paths", onglet "Librairies", et ajoutez un jar externe (celui du serveur).