Assignment 1 Preliminary Design

Java will be used in this assignment. The communication method we choose will be RMI.

**Step 1**

Server:

Client:

**Step 2**

Two threads in each peer. (actually RMI is implicitly using multi thread design, there is a main thread waiting for and dispatch client request and numerous threads dealing with client request)

1st thread is responsible for the client feature in Step1.

2rd thread deal with the maintaining job, i.e., judging server/player crashes, selecting new servers. This thread is called daemon thread. This thread has a heartbeat method, to check whether servers are alive.

Designing ideas:

1. there is one peer, whose ip address and port number are known to all peers, within 20 seconds, it act as the primary server.
2. when a peer join, the primary server remember the peer’s ip address, and allocate a unique port number to this client, the port will be used when the joined peer is selected as a server.
3. when game start, client make a movement, server response the maze map to client.
4. newly selected servers should register their services dynamically.
5. each movement will attempt to connect with both primary server & backup server, note that the two servers may not be out of use at the same time.
6. only the server is responsible for selecting next server, clients are only allowed to check the up-to-date information.
7. the first peer act as server should also create player info, and participate in the game.
8. primary server and secondary server should synchronise maze info timely. the way how to synchronise info is still yet determined.

Hints:

there exist a static method in java.rmi.server.RemoteServer：String getClientHost() — to get client’s ip address. The initial primary server will record all peers’ ip address and port number info.(port number is generated dynamically)