

```
- Game game
- ServerSocket serverSocket
- ConcurrentLinkedQueue<InputMessage> queue
+ void sendGame(Game game)
+ String receiveInput()
+ void sendResponse(Response response)
+ void run();
void sendGameToAllClient();

    intialization

serverSocket.acceptConnection()
player connect -> thread
SubThreadTask {
//when player win or lose isFinished==True
while (Player.isFinished()) {
while user not finish input {
  receive(m source des 1)
  Response game.validate(receivedOrder);
  queue.offer(receivedOrder);
  send(respond)
wait() //wait unitl notified by the Game when all orders are done
serverSocket.sendGame(Game)
while (connectedPlayer < number Player) {
accept();
while(!game.getIsEnd()){
game.startOneTurn(); //do a depp copy of current Board to validate actions
waitUnitilAllDone()
game.applyAllOrders(Queue<InputMessage>);
game.addOneToAllTerritory();
sendGameToAllClients()
```

GameServer