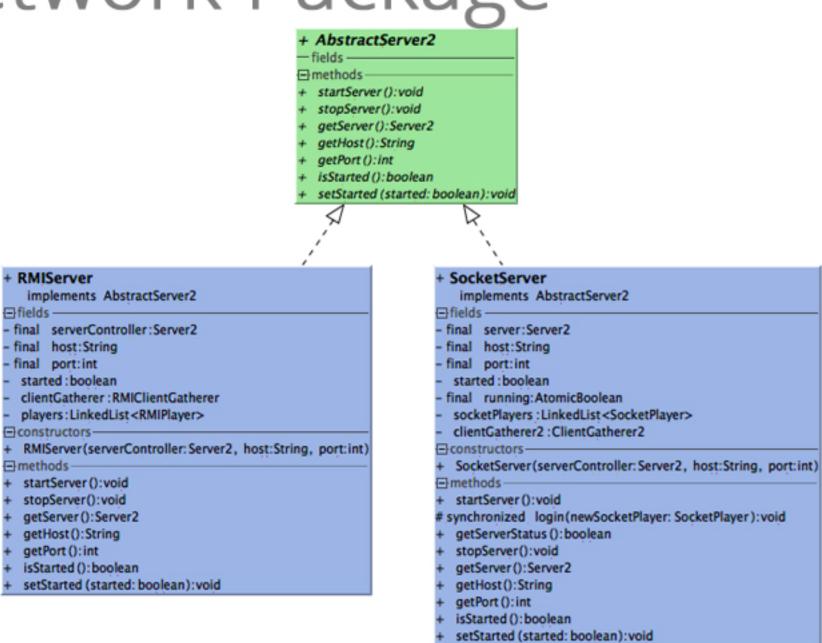
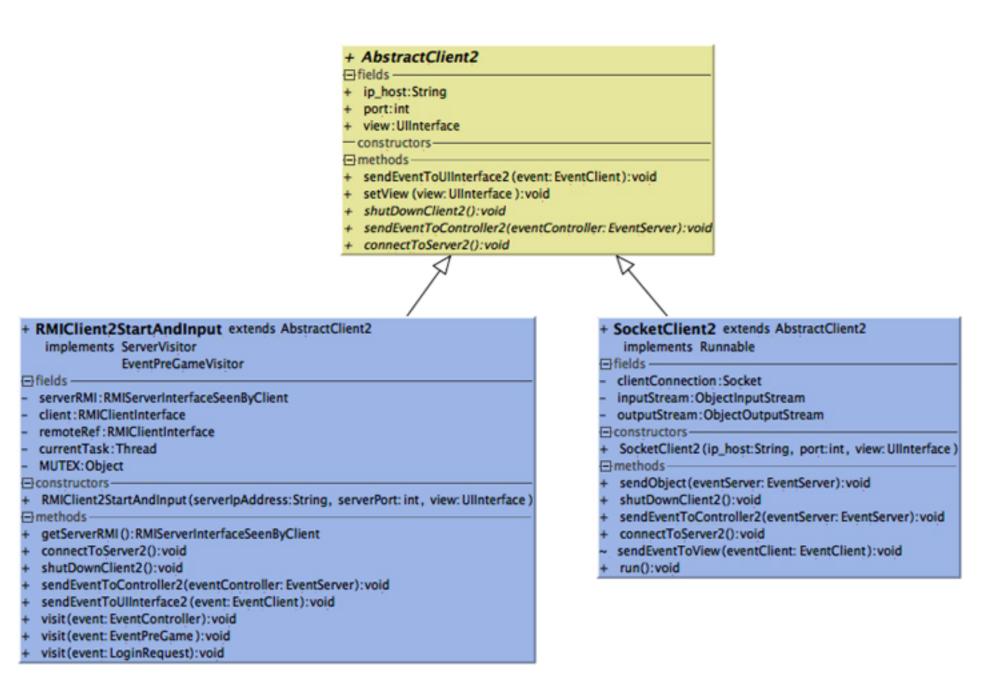
Network Package





GameInterface sendEventToGameRoom(eventController: EventController):void sendEventToView(eventClient: EventClient):void reLogin(oldRemotePlayer: RemotePlayer2, newRemotePlayer: RemotePlayer2):void disconnectFromGameRoom(oldRemotePlayer: RemotePlayer2):void + GameRoom extends Thread implements TimerCallback GameInterface players:LinkedList<RemotePlayer2> playersInGame:RemotePlayer2[] controller:Controller idGameBoard:int roomTimeout:int maxPlayer:int currentConnected:AtomicInteger timerThread:TimerThread closed:AtomicBoolean controls:LinkedList<EventController> constructors-GameRoom(idGameBoard:int) loadConfigGame():void get(index:int):RemotePlayer2 size():int synchronized startGame ():void synchronized resetOrStoreGameRoom():void addRemotePlayer (remotePlayer: RemotePlayer2):void removeRemotePlayer (remotePlayerDown: RemotePlayer2):void reLogin(oldRemotePlayer: RemotePlayer2, newRemotePlayer: RemotePlayer2):void timerCallback ():void timerCallbackWithIndex (infoToReturn:int):void sendEventToGameRoom(eventController: EventController):void sendEventToView(eventClient: EventClient):void disconnectFromGameRoom(oldRemotePlayer: RemotePlayer2):void

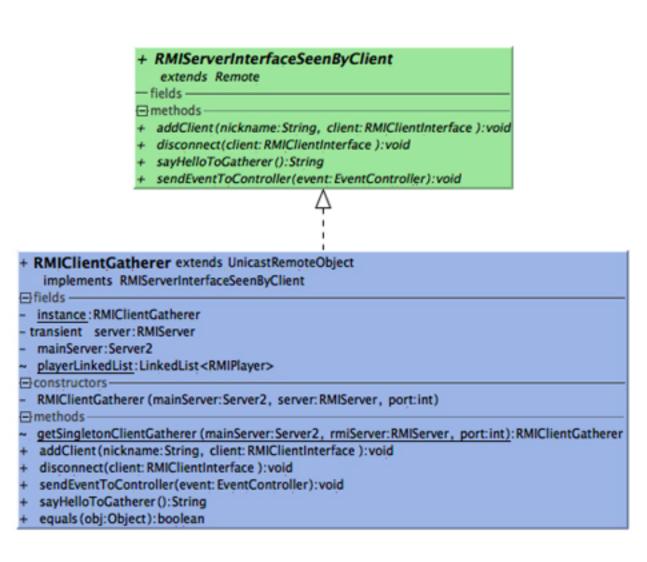
+ ClientGatherer2 extends Thread

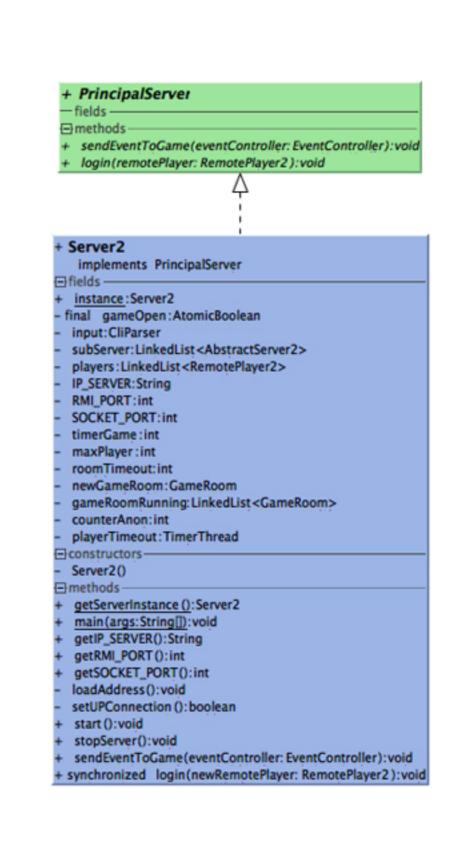
ClientGatherer2 (server: PrincipalServer, port:int)

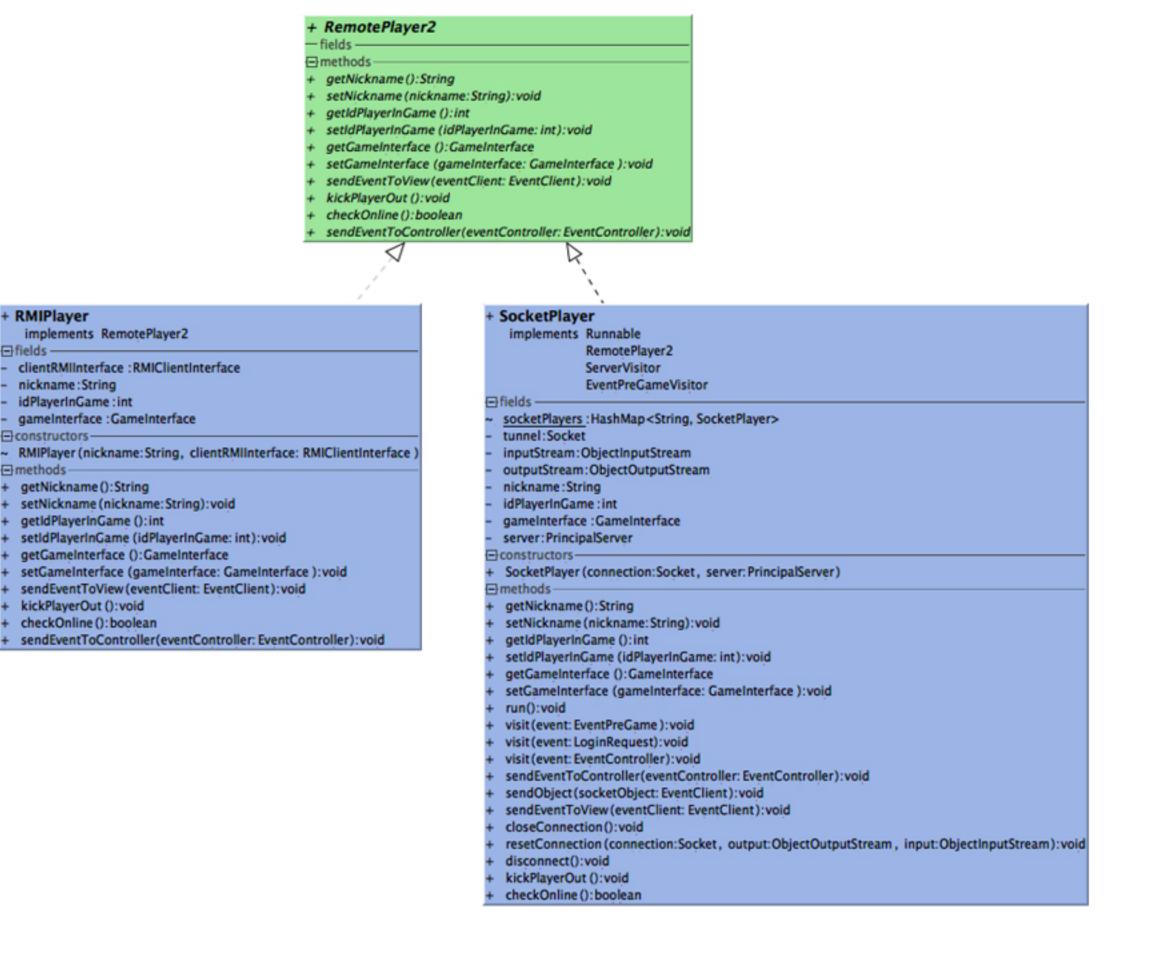
server:PrincipalServer

- final running: AtomicBoolean

serverSocket:ServerSocket









+ ClientFactory instance: ClientFactory ipServer:String rmiPort:int socketPort:int abstractClient:AbstractClient2 constructors----ClientFactory() getClientFactory (): ClientFactory loadDefault ():void createClient (view: UlInterface, serverlpAddress:String, port:int, rmi0socket1:int, cli:boolean):AbstractClient2 main(args:String[]):void

