In-class Assignment
Sockets - Chat Room

Jaap Geurts

22 November 2017

1 Introduction

In this assignment you'll build a chat room application. The assignment is divided into two parts. First you'll study a JavaFX application without any network functionality. This application shows you how to program JavaFX with a `ListView` and a `TextField`. After that you'll add the socket code to make the connections work.

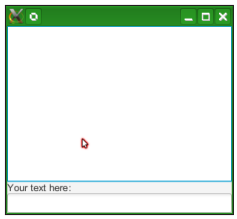

Figure 1: Chat Window

2 Provided code

You are given two Netbeans projects. The first project is named ChatLocal and contains a working JavaFX chat window implementation. It only sends text to itself. You can study it to see how JavaFX works.

The second project named Chat-Peer2Peer contains partly finished code that you'll have to extend and complete. The provided project implements the graphical user interface built in JavaFX. You have to write the missing code for making the connection (sockets) between the client and server. You'll also write the code to send messages back and forth.

To make the connection you should use TCP sockets. TCP socket connections require a client and a server to work. When you open the Netbeans Chat-Peer2Peer project you'll see that it contains the following two classes:

- `ChatServer`
  This class contains the server side of the Chat Room implementation. It also contains a `main` method to start the server.

- `ChatClient`
  This class contains the client which connects to the server. It also contains a `main` method to start the client.

3 Assignment

1. Implement the missing code to make the chat room work between the client and the server. This version only supports communication between one client and one server.

2. Extra: Modify the program such that multiple clients can connect to one server. Make sure that everyone receives all messages and no one receives duplicates.

Hints

- To test the program start the server first and then the client.
- To keep the GUI responsive use thread(s).
- To close the application nicely, i.e. without exceptions, you can interrupt the thread(s) and call `Socket.close()`.
- To prevent the client application from consuming too many CPU resources you should limit the number of connection attempts per time period between failed connection attempts.
- The client is configured to connect to the server on the same machine(localhost). If you want to test the program on two machines, you'll have to change the 'hostname' variable in the client app to the IP address of the server machine. Make sure that the server machine firewall is open for incoming connections on the listening port.