

Simple Chat Java

Elias Nachbagauer

Inhalt

Angabe.....	2
Lösung:	4
Probleme:	4

Angabe

Die aufgabenstellung war diese Angabe in Java zu realisieren:

Die Aufgabenstellung

Erstelle ein einfaches Chatprogramm auf Basis der Abbildungen unter Anwendung des MVC Design Patterns!

Grundanforderungen

GUI's mittels qt-Designer und pyqt5 [1] erstellen

Trennung jeweils von View und Controller sowie der Kommunikationsimplementierung

Sphinx-Dokumentation

Client

Die GUI soll ein Eingabefeld für Nachrichten und einen Senden-Button implementieren.

Verbindung zum Server Port 5050 wird beim Starten hergestellt - im Fehlerfall wird eine MessageBox mit dem Fehler angezeigt und das Programm wieder geschlossen.

Bei einem Klick auf "Senden" wird die Nachricht an den Server geschickt und das Textfeld geleert.

In der Grundfunktionalität ist ein anonymes Chatten möglich, wobei der Chat-Text dann jeweils ganz links beginnt.

Ein zweiter Prozess liest eingehende Nachrichten und fügt sie in den Chatbereich ein.

Wird die Verbindung geschlossen, schließt sich das Programm ohne offene Ressourcen zu hinterlassen. Das Close-Event wird abgefangen und der Socket wird sauber geschlossen.

Nachabgauer

Server

Horcht auf Port 5050 in einem Prozess auf eingehende Verbindungen. Es ist wichtig, dass `accept()` in einem eigenen Prozess aufgerufen wird, da sonst die GUI einfriert.

Im Fehlerfall wird eine entsprechende `MessageBox` angezeigt und das Programm beendet.

Neue Nachrichten an den Server werden an alle Clients verteilt und im eigenen Chat Feld angezeigt.

Wenn der Server geschlossen wird, wird die Verbindung zu allen Clients sauber beendet und die Clients werden daher automatisch geschlossen.

Achtung: GUI-spezifische Operationen (z.B. Anzeigen einer `MessageBox` oder Bearbeiten eines GUI-Elements) dürfen nur vom GUI-Prozess durchgeführt werden - verwende hierfür eigene Signals und Slots.

Erweiterungen

Der User kann den Chatnamen bei Programmaufruf mitgeben.

`python MyChatClient.py ChatName`

Ohne Chatname soll in der erweiterten Ausführung jeder anonyme User einen Chatnamen automatisch erhalten (z.B. "Client 1", "Client 2"), wobei der chat Text dann jeweils nach dem Doppelpunkt beginnt. Es wird immer "ChatName: text" gesendet, also "Client 1: Hallo!", "Franz: Servus!".

Neue Clients werden der Client-Liste hinzugefügt, sobald sie sich verbinden. Für jeden neuen Client wird ein neuer Prozess erstellt, welcher auf eingehende Nachrichten wartet. Wenn ein Client die Verbindung beendet, wird er aus der Client-Liste entfernt.

Der ServerAdmin kann einen Client entfernen indem er ihn in der `UserListBox` anklickt und dann den `Disconnect Button` drückt. Der Server kann also verbundene Clients aus der Client-Liste entfernen und die Verbindung zu ihnen beenden.

Der Client muss konfiguriert werden, bevor er die Verbindung mit dem Server herstellt: Parameter sind ChatName, Host, IP.

Nachabgauer

Im Falle einer langen Chat Historie soll die Chat List box automatisch ganz nach unten scrollen damit man bequem mitlesen kann.

Mach dir außerdem Gedanken über Prozesssynchronisation - muss etwas synchronisiert werden?
Wenn ja, was? Halte deine Erkenntnisse hier fest!

GUI skaliert bei einem Resize entsprechend.

Lösung:

Um diese Aufgabe zu lösen habe ich mir das Tutorial (<https://www.youtube.com/watch?v=DbpFZP3ITys&list=PLE4E8B5A20B0A5091>) angesehen und aus dem von dieser Videoreihe entstanden Code die Fehlenden Aufgaben gelöst. Zu diesem Zweck habe ich in das bestehende Programm mit einer Comand-Funktion erweitert damit man die Nutzerliste („Map“) anzufordern. Diese Liste wird beim Anmelden von Clients um den Client erweitert und beim Abmelden und den Client wieder Reduziert. Die zweite Funktion die ich Implementieren musste war das saubere Schließen welches ich auch über das Comand gelöst habe („/Exit“) oder durch das schließen des Fensters ausgelöst wird. In diesem Fall wird der zu dem Client gehörige Socket geschlossen und der Thread beendet. Die genaue funktionsweise wird im Tutorial sehr gut und verständlich erklärt.

Probleme:

Das größte Problem war das saubere beenden, welche ich aber nach einigem Nachschauen in der API und überlegen behoben werde konnte.