Institut für Informatik
Lehr- und Forschungseinheit für Theoretische Informatik
Dr. Thorsten Altenkirch, Dr. Ralph Matthes
Thomas Anderl, Alessandro Macri

Programmierpraktikum
WS 1999/00

1. Milestone (Siedler von Catan)

10. 11. 1999

Deadline: 24.11.1999

Als erstes soll ein Chatsystem implementiert werden, mit dem während des Spiels Verhandlungen geführt werden können. Es wird ein Chatserver realisiert, mit dem die Chatklienten der Spieler über `Sockets` kommunizieren.

In der ersten Ausbaustufe weist der Server den (max. 4) Klienten zufällig Farben zu (nämlich die Farben des Spiels, wobei es sinnvoll ist, weiß durch gelb zu substituieren), wobei natürlich jede Farbe nur einmal vergeben wird. Der Server teilt dem Klienten zunächst seine Farbe mit, danach wartet er auf Eingaben der Klienten (in Form von Zeilen), die er dann an alle Klienten weiterleitet. Das Protokoll soll so definiert werden, daß der Server den Klienten die Farbe des Senders der Zeile mitteilt.

Der Klient soll über ein einfaches graphisches Interface bedienbar sein, daß mit Klassen aus `javax.swing` realisiert wird. Sobald er vom Server seine Farbe mitgeteilt bekommt, zeigt er diese an (z.B. als Hintergrundfarbe). Über ein `JTextField` kann der Benutzer eine Zeile eingeben, die dann über den Server an alle Klienten weitergeleitet wird und dort in einer scrollbaren Komponente verschiedenfarbig angezeigt wird.

Die wesentlichen Konzepte von Client-Server-Programmen sind sehr gut im *Java Tutorial* in der Lesson *All About Sockets* des Trails *Custom Networking* beschrieben. Ein geigneter *Port* für unseren Chat-Klienten ist z.B. 4000.

Da der Server mit mehreren Klienten Verbindung halten muß, ist es zweckmäßig, `Threads` zu verwenden. Der Server muß also für jeden Klienten einen neuen Thread starten. Es ist wichtig, beim Zugriff auf gemeinsame Daten `synchronized methods` zu verwenden, um inkonsistente Zustände zu vermeiden.

`Threads` werden imTutorial in der Lesson *Doing Two or More Tasks At Once: Threads* des Trails *Essential Java Classes* beschrieben.

Es ist empfehlenswert, sich zunächst auf das Protokoll zu konzentrieren und ein Minimalinterface auf Textbasis zu implementieren.

Dann ist es sinnvoll sich mit dem Swing API auseinanderzusetzen. Zu diesem Thema gibt es den Trail *Creating a GUI with JFC/Swing* im *Java Tutorial*.

Eine sinnvolle Erweiterung ist, den Spielern eine bevorzugte Farbe angeben zu lassen. Ist die bevorzugte Farbe noch nicht vergeben, wird sie dem Spieler zugewiesen, ansonsten erhält er eine zufällige Farbe.