

Softwareentwicklungspraktikum  
SS 2019

Übungsblatt 1: Milestone I – Love Letter – Netzwerk

Als erstes, kleines Projekt nehmen wir uns das Kartenspiel “Love Letter” des japanische Spieleautors Seiji Kanai vor. Im Groben geht es in diesem Spiel darum, (möglichst viele) Liebesbriefe an eine Prinzessin zu senden, um sich so um ihre Gunst zu bemühen.

Verschiedene Persönlichkeiten am königlichen Hof streben dieses Ziel an, wobei sie, basierend auf ihrem Stand und ihren Fähigkeiten, unterschiedlich gute Chancen haben. Die Regeln können im Detail unter [http://alderac.com/wp-content/uploads/2017/11/Love-Letter-Premium\\_Rulebook.pdf](http://alderac.com/wp-content/uploads/2017/11/Love-Letter-Premium_Rulebook.pdf) (beachten Sie, dass die Spielversion mit 2–4 Spielern verwendet werden soll) eingesehen werden.



Zu diesem Projekt gibt es drei Aufgabenblätter (1–3). Ihre wöchentlichen Fortschritte zeigen und besprechen Sie bitte mit Ihrem Tutor. Die Abnahme für dieses Vorprojekt (über die Anforderungen aus allen drei Aufgabenblättern) wird voraussichtlich zwischen 20.05. und 24.05. stattfinden.

Allgemeine Hinweise:

- Es ist für diese Veranstaltung essentiell, dass Sie lernen sich Kenntnisse *selbstständig* anzueignen. Dazu zählt insbesondere die Bedienung einer Entwicklungsumgebung (z.B. Eclipse, IntelliJ IDEA, etc.) und der Versionsverwaltung Git.

- Treffen Sie sich regelmäßig (mehrmals pro Woche) mit Ihren Teammitgliedern!  
Es wird notwendig sein, dass Sie als Gruppe arbeiten und das Projekt nicht vollständig aufteilen. Arbeiten Sie als Gruppe zusammen, kommunizieren Sie und nutzen Sie die Versionsverwaltung Git (Pflicht!). Dazu müssen Sie sich gegebenenfalls im Kennungskonfigurator CipConf (<https://tools.rz.ifi.lmu.de/cipconf/>) eine Gitlab2 Kennung anlegen. Erledigen Sie das vor dem ersten Tutorentreffen, damit Sie von ihrem Tutor zu ihrem Projekt hinzugefügt werden können. Die Projekte aus dem Softwareentwicklungspraktikum sind Gruppenarbeiten und *jedes Gruppenmitglied* soll dazu beitragen!
- Achten Sie darauf, dass alle Anwendungen, die Sie im Laufe des Softwareentwicklungspraktikums entwickeln, auch auf den Rechnern des CIP-Pools ausführbar sind und korrekt laufen. Die Abnahmen werden auf diesen Rechnern stattfinden! Um auf dem aktuellen Stand arbeiten zu können, stellen wir Ihnen eine Anleitung zur Einrichtung von Java 11 und JavaFX 11 bereit.

### Aufgaben:

- Vereinbaren Sie und Ihre Gruppe einen wöchentlichen Termin mit dem Ihnen zugeteiltem Tutor.
- Installieren Sie eine Entwicklungsumgebung und das Git-Plugin.
- Zur Implementierung der grafischen Oberfläche muss JavaFX verwendet werden.
- Initialisieren Sie Ihr Projekt (z.B. die Projektvorlage) in ihrem Gruppenverzeichnis in Git, d.h. in der Adresse [https://gitlab2.cip.ifi.lmu.de/dbs\\_sep2019/<GRUPPE>](https://gitlab2.cip.ifi.lmu.de/dbs_sep2019/<GRUPPE>). Sie werden von ihrem Tutor im ersten Treffen hinzugefügt.
- Sie können die Projektvorlage von Git herunterladen und diese als Grundlage für ihre Implementierung verwenden.
- Beginnen Sie die Implementierung ihres Spiels “Love Letter” mit einem Chatsystem über TCP, bei dem sich beliebig viele Clients bei einem Server anmelden können.
  - Bei der Anmeldung muss der Benutzer im Client zu Identifikationszwecken einen Nicknamen *name* angeben, den der Server aber nur akzeptiert, wenn *name* aktuell nicht vergeben ist.
  - Nach erfolgreicher Anmeldung wird der neue Teilnehmer in seinem eigenen Ausgabefenster vom Server begrüßt (z.B. “welcome *name*”), die übrigen Clients bekommen in ihrem Ausgabefenster eine Meldung, dass der Benutzer beigetreten ist (z.B. “*name* joined the room”)
  - Das Ausgabefenster, das jeder Client zur Verfügung stellt, dient zur Ausgabe sämtlicher Nachrichten, insbesondere derjenigen, die von den anderen Teilnehmern kommen. Die Ausgabe in diesem Fenster muss asynchron erfolgen, d.h., auch wenn sich der entsprechende Benutzer gerade beim Tippen eigener Nachrichten befindet.
  - Sobald ein Client in seinem Eingabefenster eine Zeile eingegeben hat, muss bei allen angemeldeten Clients diese Botschaft (asynchron) im Ausgabefenster erscheinen.
  - Mittels Eingabe von “bye” wird die Verbindung zum Server beendet, alle anderen Teilnehmer erhalten in ihrem Ausgabefenster (wieder asynchron) die Nachricht “*name* left the room”.
  - *Optional: Wenn Sie ihre Nachrichten serialisieren wollen können Sie die dafür benötigten Bibliotheken verwenden!*
- Sie haben für dieses Übungsblatt zwei Wochen Zeit, für die darauffolgenden Übungsblätter dagegen nur noch jeweils eine Woche.

### Hinweise zur Implementierung während des gesamten Praktikums:

- Achten Sie stets auf die Trennung von Model, View und Viewmodel (MVVM) und auf eine modulare und erweiterbare Entwicklung ihres Spieles.
- Im Laufe des Vorprojekts müssen Sie den Chat erweitern und anhand dessen das Spiel modellieren. Überlegen Sie sich *gemeinsam* in ihrer Gruppe ein Daten-Modell für das Spiel “Love Letter”. Welche Objekte benötigen Sie? Wie interagieren diese miteinander? Wie müssen Client und Server kommunizieren?
- Identifizieren Sie Teile, die Sie gemeinsam implementieren müssen, und Teile, die Sie unabhängig implementieren können.
- Speichern Sie ihre Änderungen *oft* in Git. Lernen Sie, wie man Bearbeitungskonflikte dabei auflöst und alte Versionen wiederherstellt.