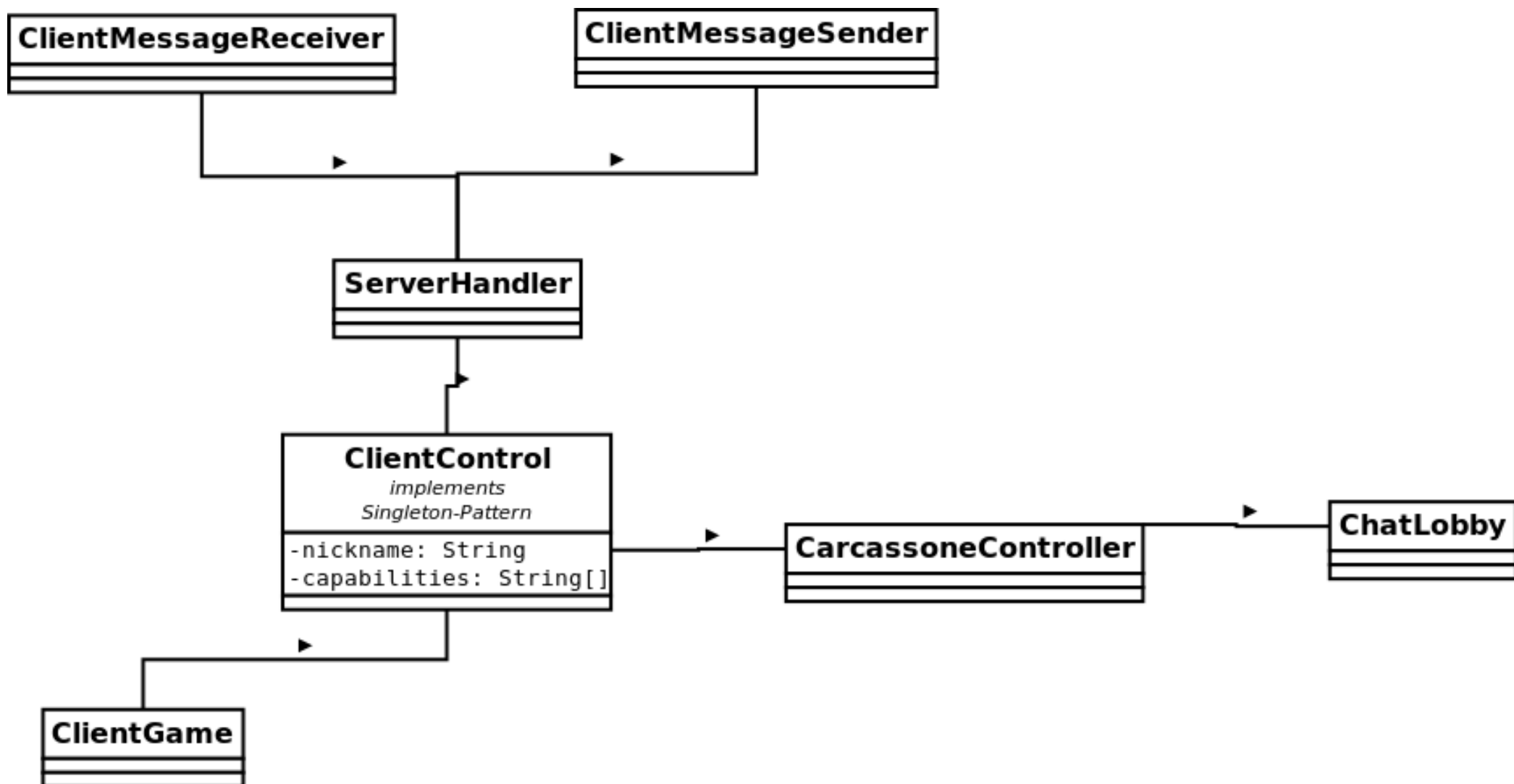


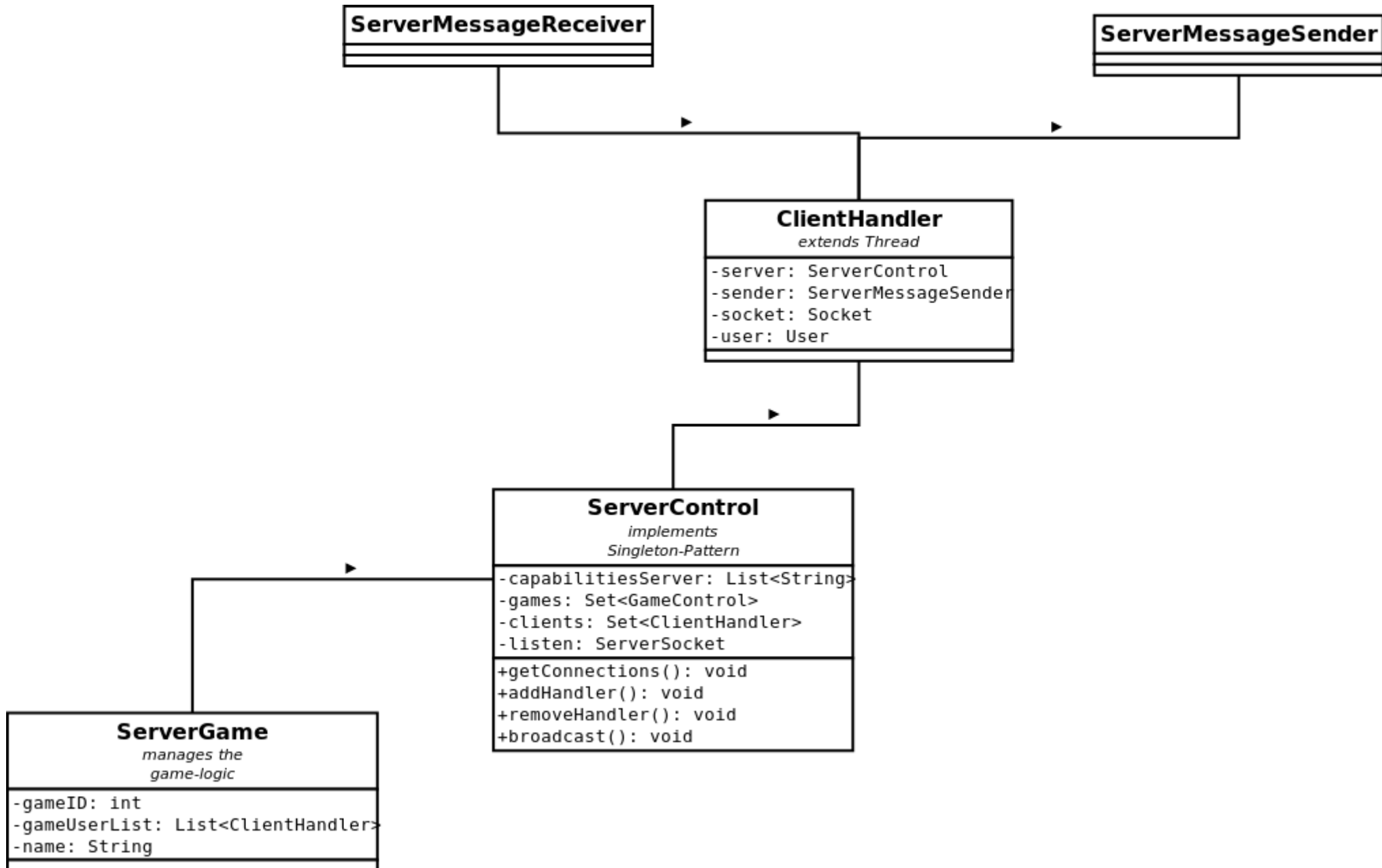
Team Lannister

1. Organisation der Gruppe (Matthias)
 - 1.1 Untergruppen
 - 1.2 Gruppentreffen
 - 1.3 Zeitplanung
 - 1.4 Probleme
2. Aufbau der Software
 - 2.1 Demonstration (Victoria)
 - 2.2 UML Allgemein (Matthias)
 - 2.3 Einzelne Bestandteile der Software
 - 2.3.1 View (Ramona)
 - 2.3.2 Server/Client (Michael)
 - 2.3.3 Logik (Jonny/Jonas)

Organisation

- Model/Logik: Johannes & Jonas
 - Server/Client: Matthias & Michael
 - View: Ramona & Victoria
-
- 25.11: Erzeugen von Spielen und beitreten
 - 2.12: Gamechat funktioniert, Überprüfung ob Zug korrekt
 - 9.12: Meeplesetzung, Berechnung der Punkte teilweise
 - 16.12: Prototypfertig

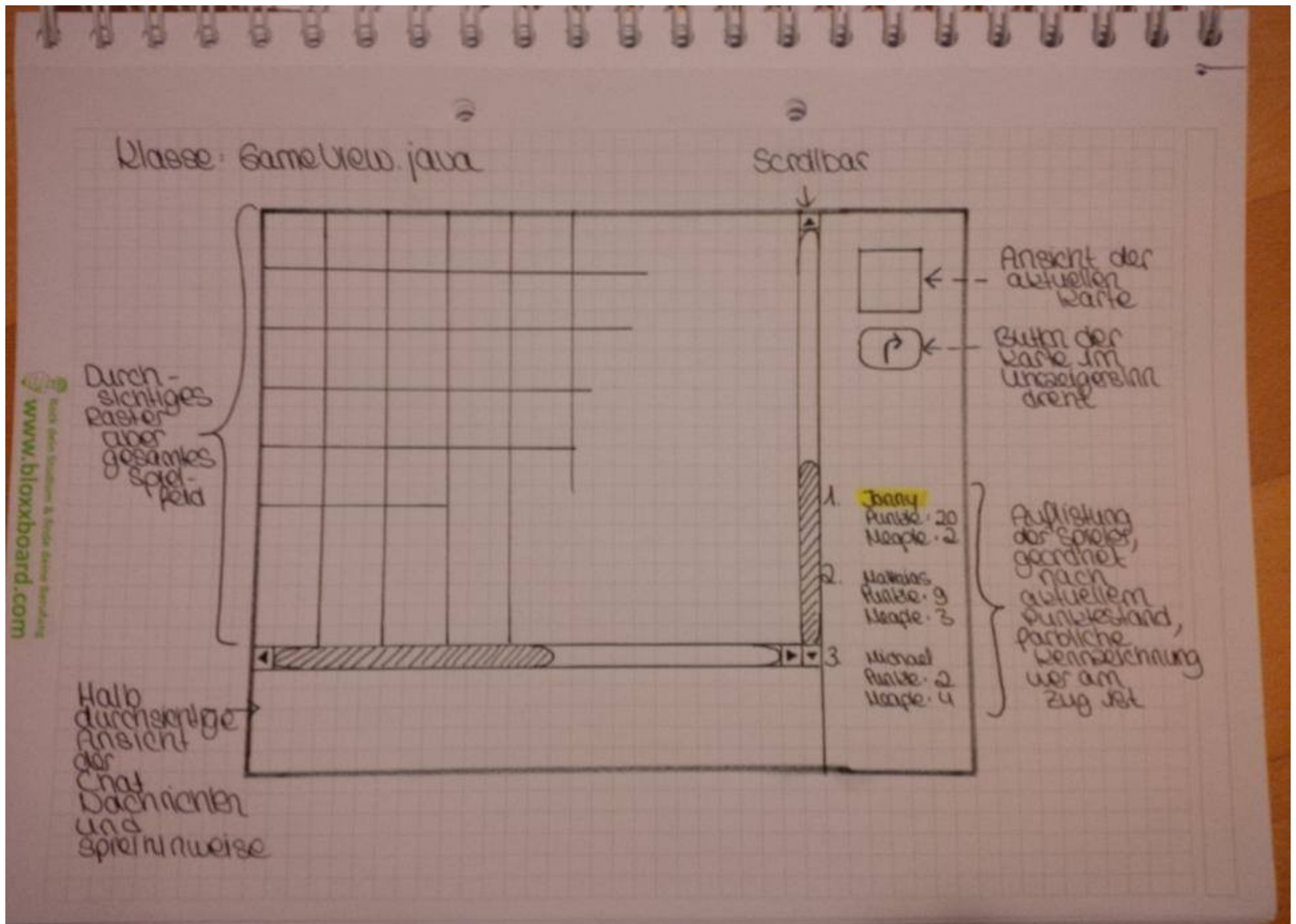




Warum JavaFX?

- Relativ neu
- Schöner als Swing (kein veraltetes Design)
- Stylesheets können direkt eingebunden werden
- Insgesamt einfacher zu lernen und handzuhaben
- Problem: nicht so viele Informationen als Hilfe im Internet

GameView



Aktuelle GameView



Player

Kein Content in Tabelle

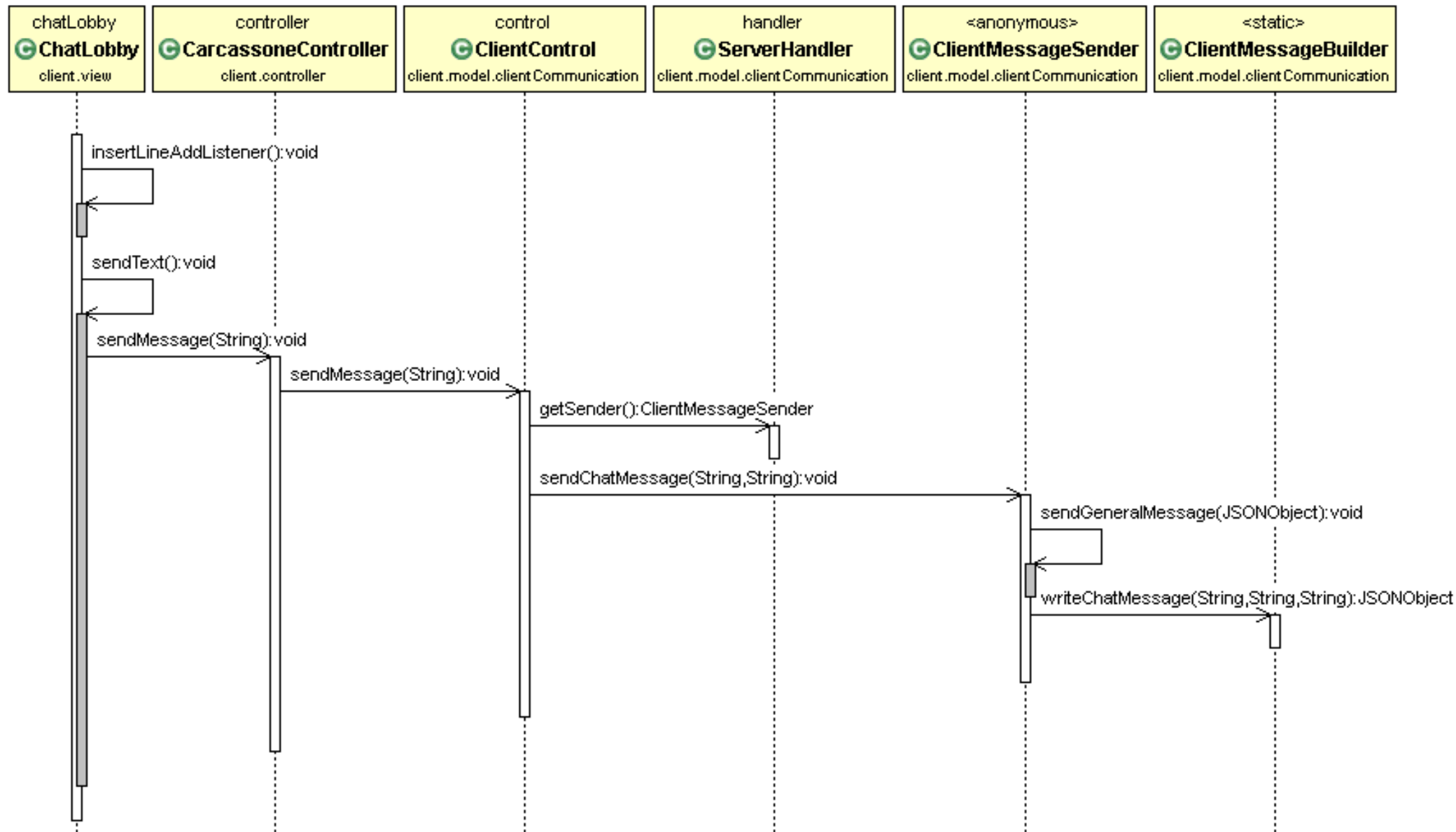
03:37:22 r: hi

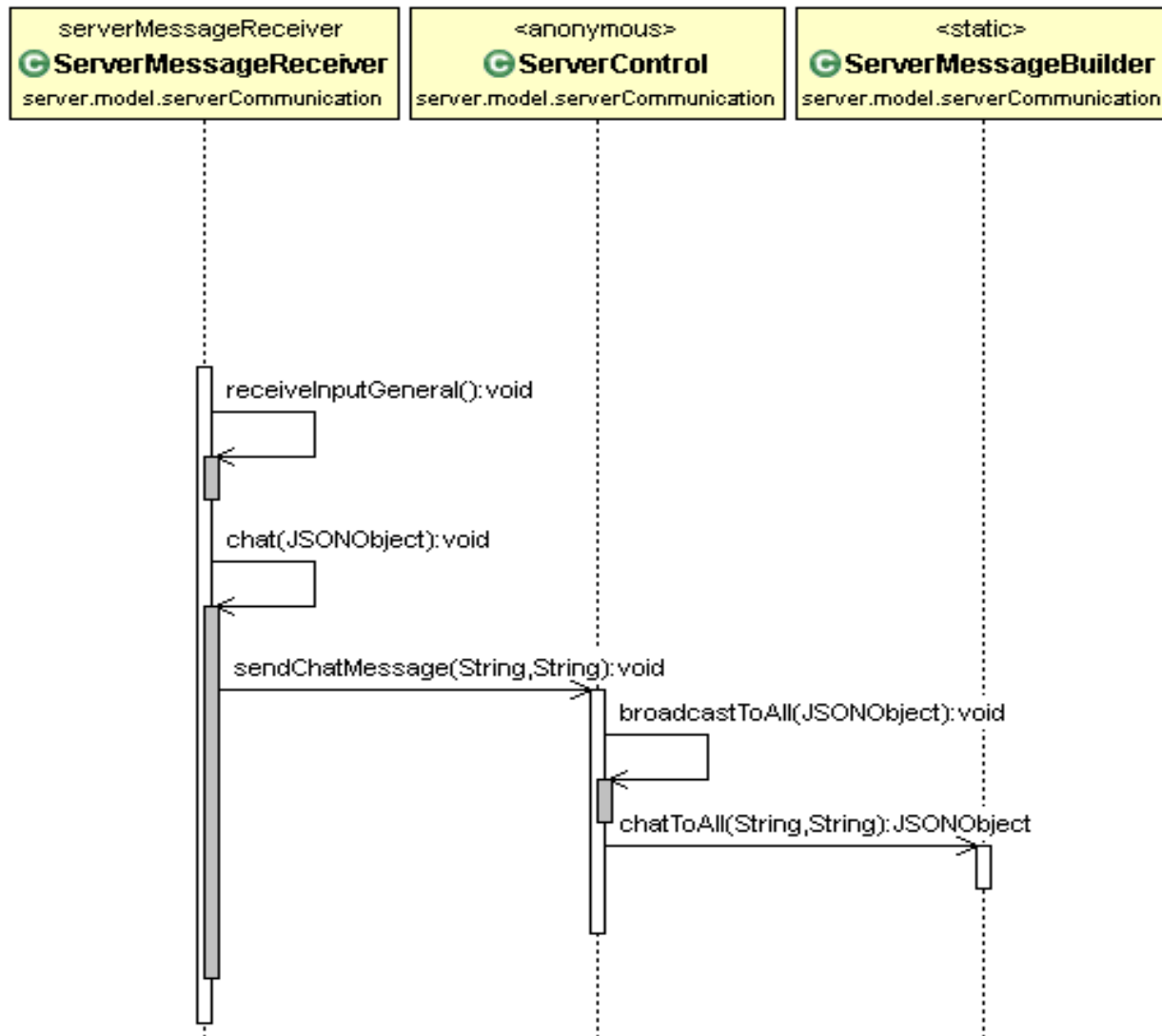
Send

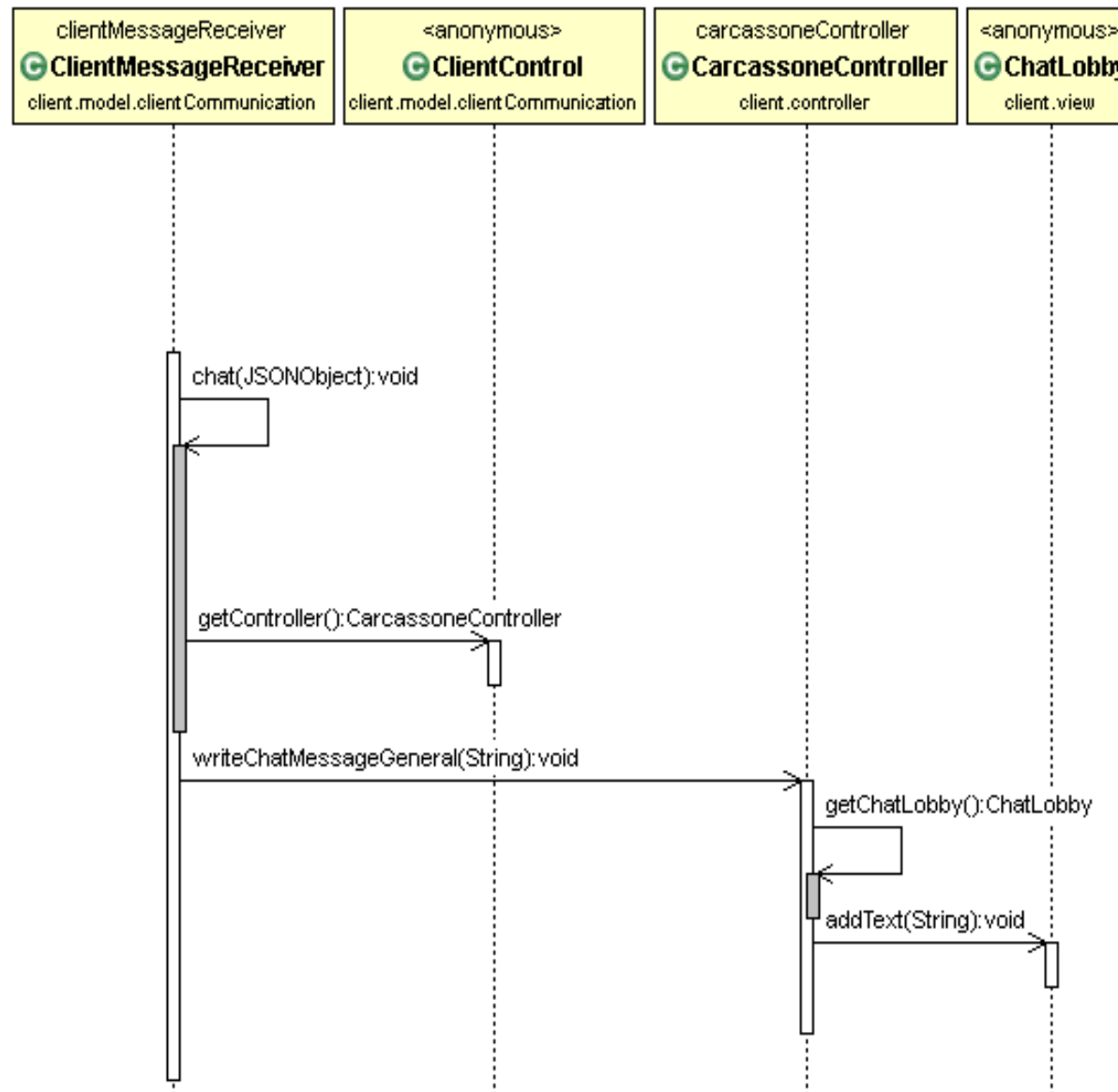
Allgemeine Server/Client Architektur

- Server multithreaded
- MessageBuilder (utility class) zur Kapselung
- Enums zur Verbesserung der Typsicherheit
- Singleton

Chat-Sequenzdiagramm







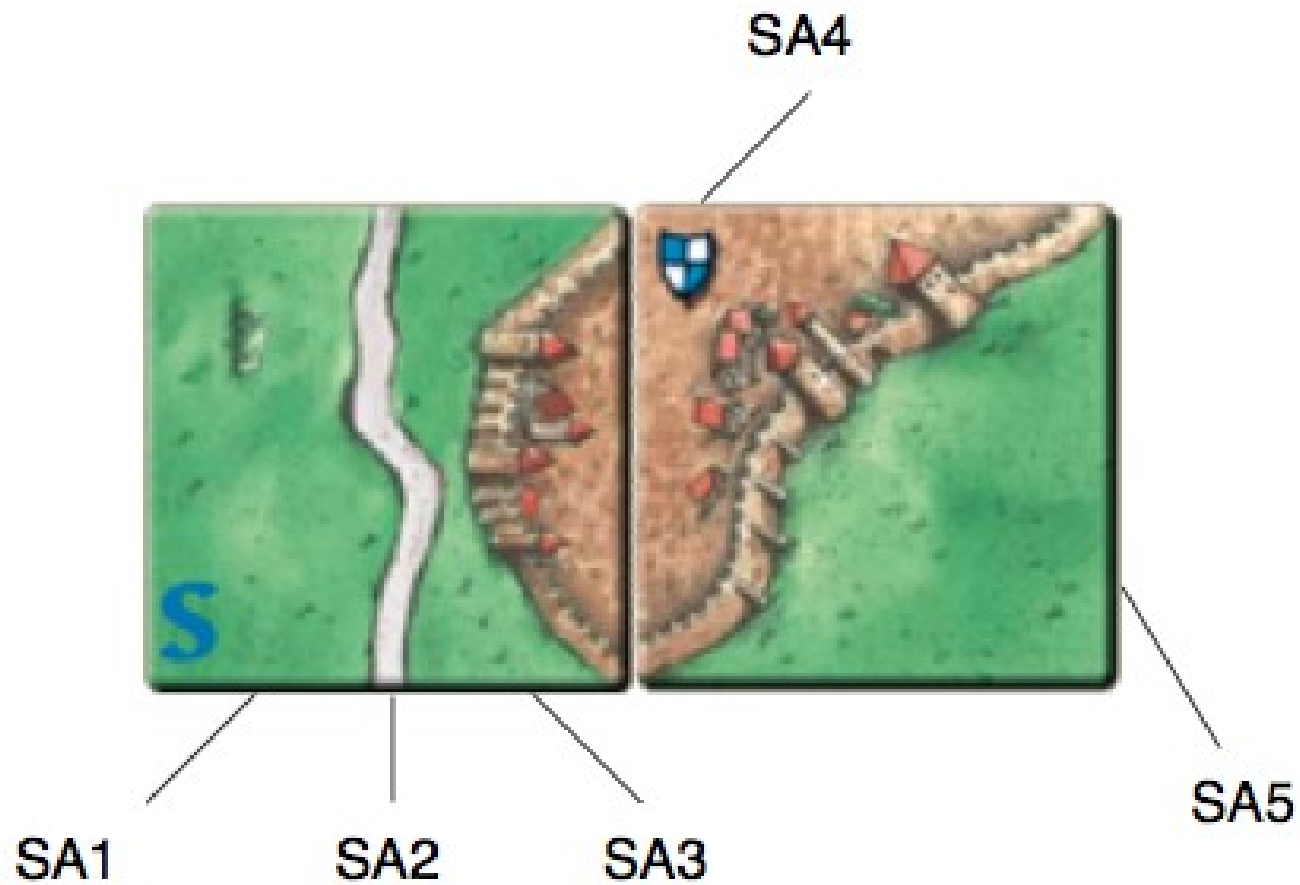
JSON-Messages

- TODO:
 - “game removed” , wenn Stack.size() mit Karten == 0
 - “move”, Logik und View Methoden fehlen
 - “move failed”, Logik und View Methoden fehlen
 - “tile drawn”, Logik und View fehlen
- DONE:
 - Alle anderen messages sind implementiert und funktionieren

Logik-Klassen

- Card
- LogicCardCreator
- CardDeck
- GameField

Einfärben zugehöriger areas



Auswertung

- Road
- Meadow
- Town
- Monastery