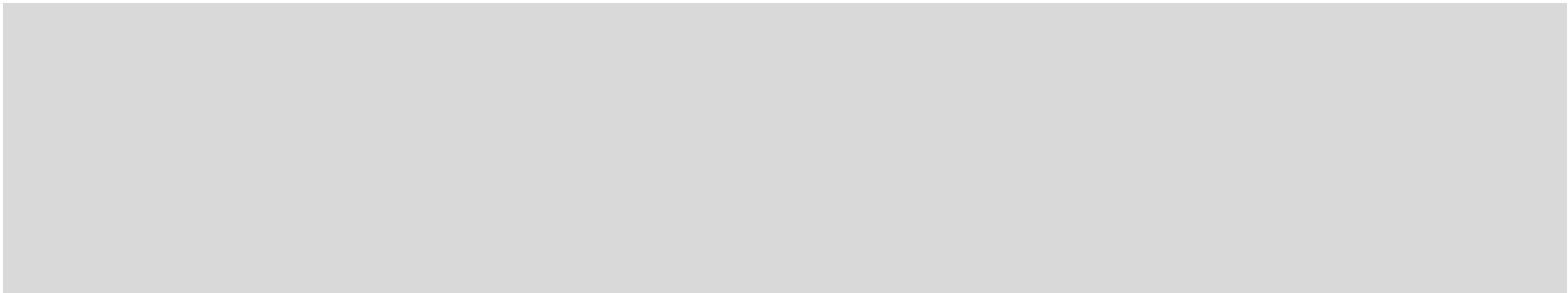




# Quasar-Beispiel: NIM

---

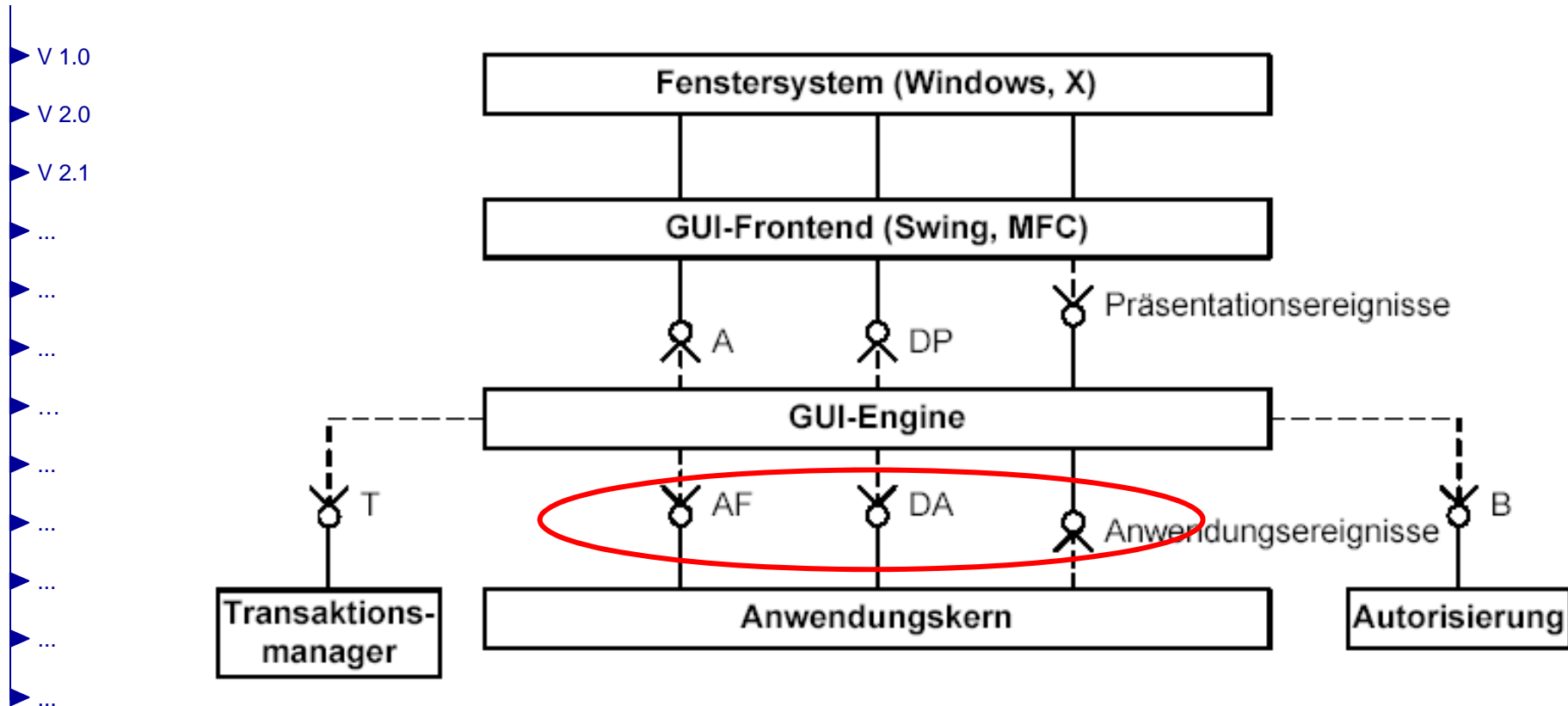
- Aufgabenstellung





# Architektur grafischer Bedienoberflächen: Übersicht [sehr frei nach Siedersleben]

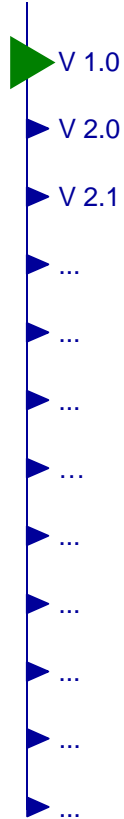
Quasar-Beispiel:  
NIM





# Aufgabenstellung: NIM-Design und -Implementierung

Quasar-Beispiel:  
NIM



Gegeben ist die Beschreibung für folgendes Spiel

- Zunächst wird eine (beliebige, aber feste) Anzahl (z.B. 20) Hölzchen in eine Reihe gelegt
- Zwei Personen nehmen immer abwechselnd mindestens ein, höchstens drei Hölzchen
- Wer das letzte Hölzchen nehmen *muss*, hat verloren
- Jeder Spieler hat jederzeit die Möglichkeit, das Spiel abzubrechen oder ein neues zu beginnen
- Die Spielkomponente merkt sich, wie oft jeder Spieler gewinnt und liefert auf Anfrage einen einfachen High Score
- Entwickeln Sie ein Design (auf Papier) nach Quasar mit der vorgestellten Struktur für obige Aufgabenstellung:
  - Was erledigt der Server, welche Anwendungsfälle/Zustandsabfragen gibt es?
  - Was erledigt die GUI-Engine, welche Anwendungsereignisse gibt es?
- Implementieren Sie Ihr Design



# Anleitung zum *Vorgehen* in 4er- bis 6er-Gruppen

- Definieren Sie die *Interfaces*, die Sie benötigen
- Implementieren Sie arbeitsteilig jeweils zu (mindestens) zweit die folgenden Komponenten

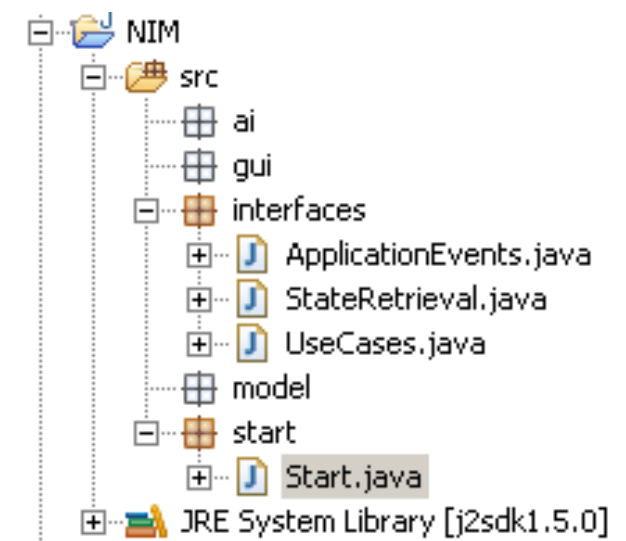
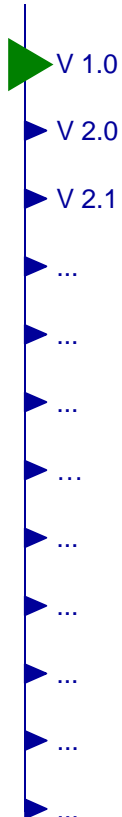
- Eine GUI-Komponente (Client) für die Interaktion mit *einem* Benutzer
- Eine Logik-Komponente (= Anwendungskern, Server)
- Eine KI-Komponente (= Computer-Spieler, Client; nur für 6er-Gruppen)

Die Implementierung findet parallel an 2 - 3 Rechnern statt

- 1 Komponente = 1 Java-Package
- Das zuerst erstellte Package *interfaces* wird auf jeden Rechner kopiert

- Eine *main*-Methode, die außerhalb der Komponenten in einem 4. Package liegt, kreiert Spieler und Modell und verbindet sie mittels geeigneter *setInterface( ... )* – Methoden (z.B. *setUseCases(...);*) dann startet sie das Spiel (eigentlich: ... beide Spieler!)

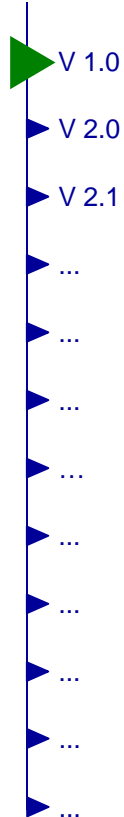
Quasar-Beispiel:  
NIM





# Anleitung zum Entwickeln der Interfaces

Quasar-Beispiel:  
NIM



- Welche Interaktionen (*Use Cases*, Anwendungsfälle) gibt es mit dem Spiel?
- Welche (groben) Abläufe (→ Aktivitätsdiagramme mit Partitions oder Sequenzdiagramme), welche Abfragen, welche Anwendungsereignisse ergeben sich daraus?
- Welche Aufgaben(teile) übernehmen welche Komponenten?
- Welche *Schnittstellen* werden dafür benötigt?
- Definieren Sie *für jede Schnittstelle ein Java-Interface*; alle Interfaces kommen in ein Package namens *interfaces*
- Erstellen Sie eine Architektur-Darstellung für Ihre tatsächliche Implementierung, welche die Komponenten und alle relevanten Schnittstellen enthält!
- *Die Herausforderung:*  
Ihre getrennt entwickelten Komponenten sollen *sofort* und *ohne Änderungen* zusammen arbeiten können!