PTP-Projekt: Schach

VON JEREMY ZAY UND SEBASTIAN FROMM

Codestruktur

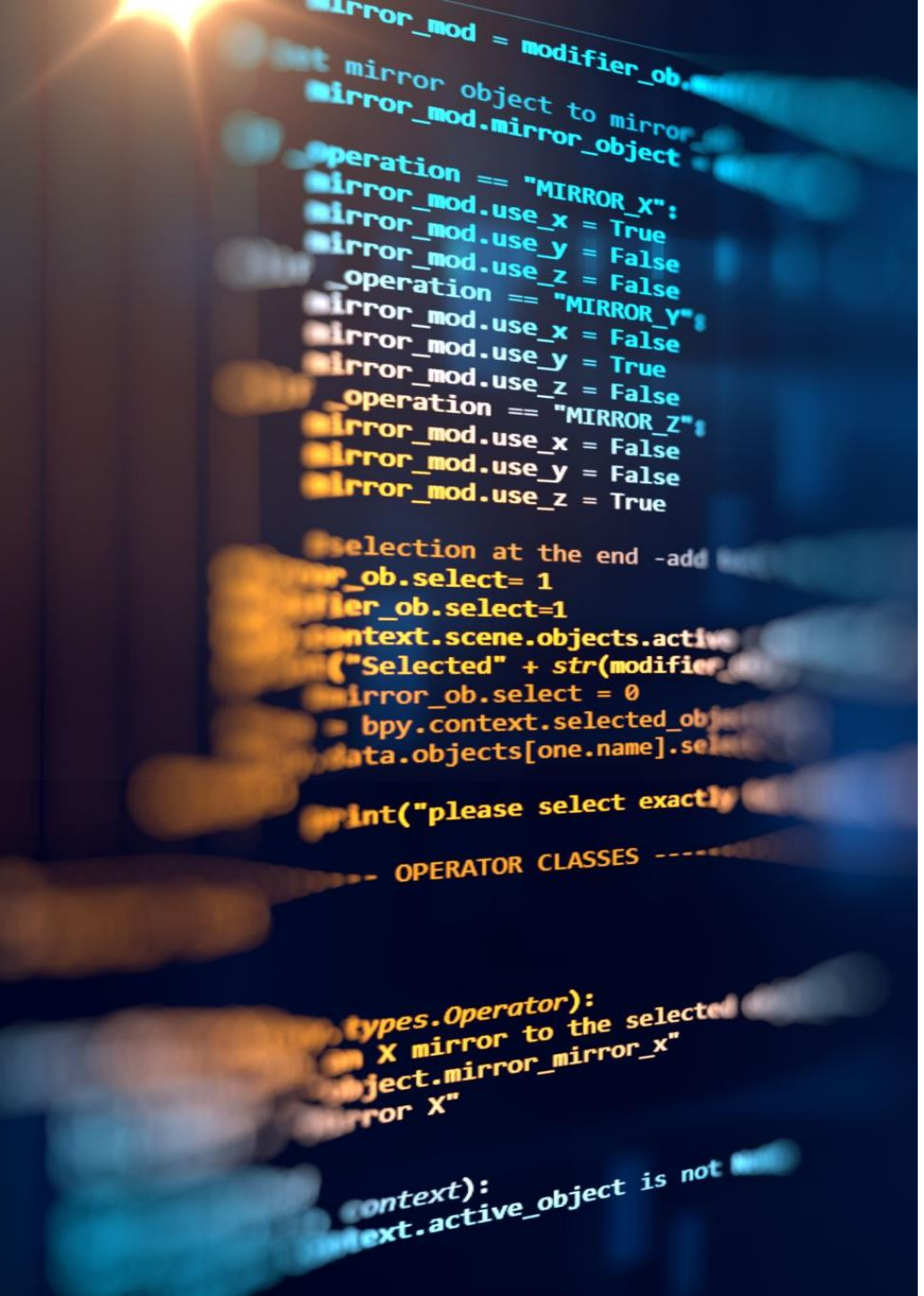
basiert auf Model-View-Controller (MVC)

Model – Spiellogik

View – UI (Java Swing)

Controller – Spielablauf & Funktionen (Save/Load Game)

Extras (z.B. Network)



Multiplayer (Peer-to-Peer)

- Spieler 1 = Host | Spieler 2 = Client
- Java Socket-API (TCP)
- Host-Client-Kommunikation über Strings (mit den relevanten Spielzuginfos)
- Spieler 1 (Host) wartet auf Verbindung von Spieler 2 (Client) durch Eingabe der Host-IP
- Probleme: Sicherheit (offener Port benötigt, wenn nicht im gleichen Netzwerk)
- Lösung: Implementation einer Verschlüsselung (z.B. TLS)

Schwierigkeiten & Selbstkritik

- paralleles Arbeiten (git Probleme, Merge)
- am Anfang selbst überschätzt

Model-Struktur

- Board: implementiert mit 1D array[64]
- OOP: Inheritance: Piece
- API: GameState

Model-Struktur

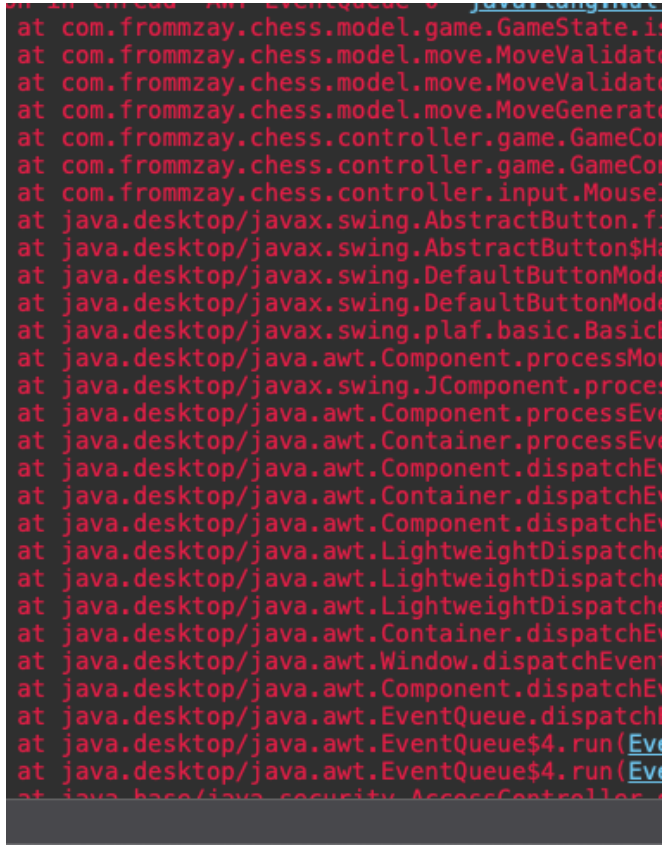
- Ziel: einfach, die Valid Moves zu finden
- 1. PseudoValidMoves
  - "King Sicherheit" wird ignoriert ❌
- 2. ValidMoves
  - "King Sicherheit" ✅
  - "Besondere" Moves (Castling, En-Passant)

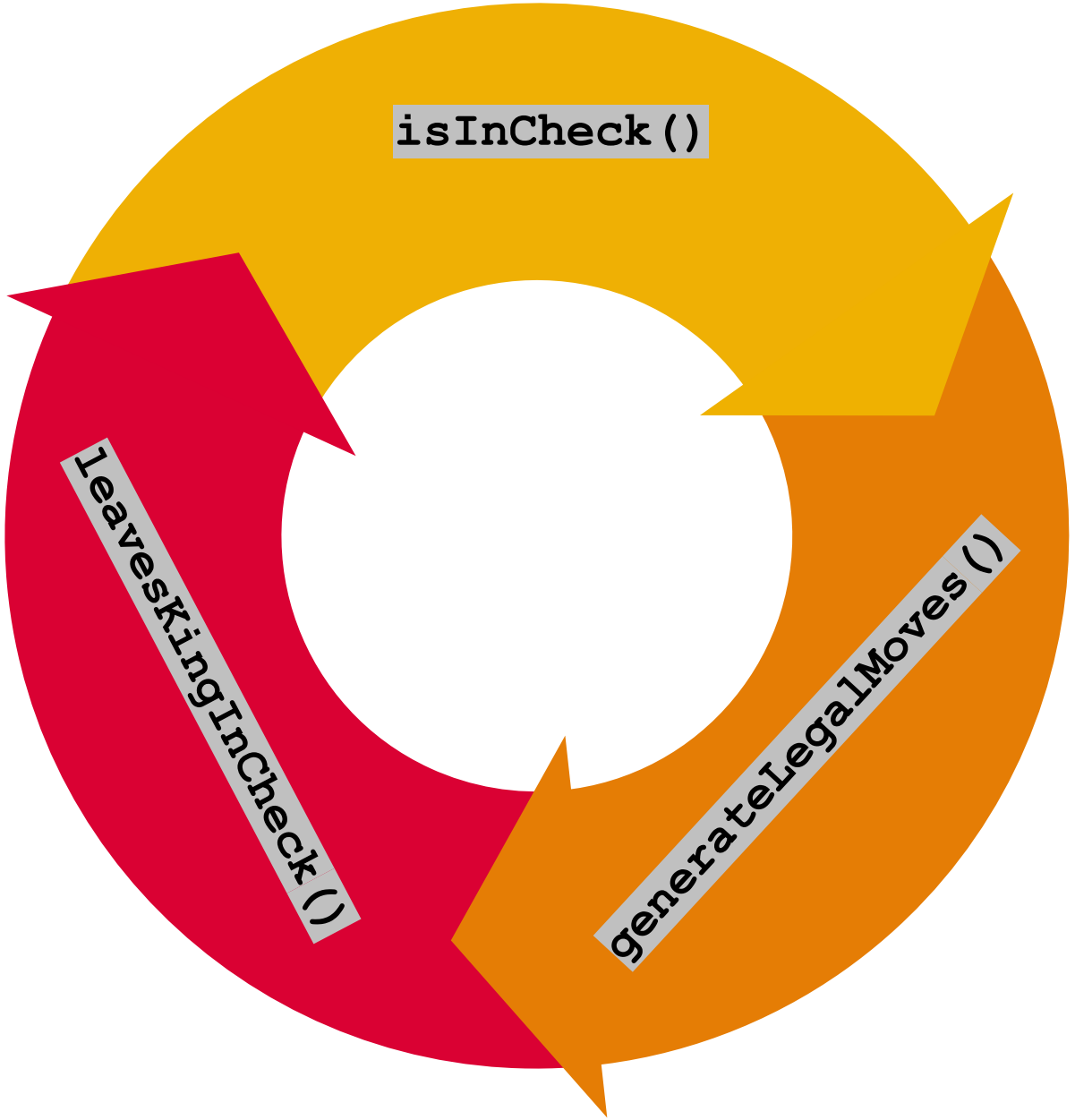
Schwerigkeiten

- Um zu wissen, ob deiner König *“in check”* nach deine “move” ist, muss man im Zukunft gucken.

```
probe = gameState.copy()
probe.applyMove(move)
illegal = probe.isInCheck()
return illegal
```

Unendliches Rekursion





Aktuelle Version