



FH MÜNSTER
University of Applied Sciences

Softwareentwurf

Praktikum WS 2025/26

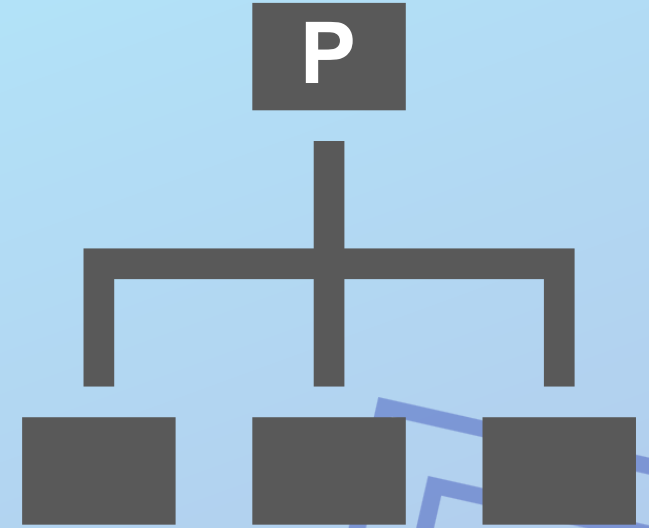
FHMaze

Prof. Dr. Patrick Stalljohann

Labor für Softwarearchitektur

Stegerwaldstraße 39 fon +49 (0)2551-962211
D-48565 Steinfurt Raum D 223 / A 212

stalljohann@fh-muenster.de
<https://www.fh-muenster.de/de/eti/labore-institute/softwarearchitektur/>



STORY – MOTIVATION

Wo bin ich, was mache ich und warum?

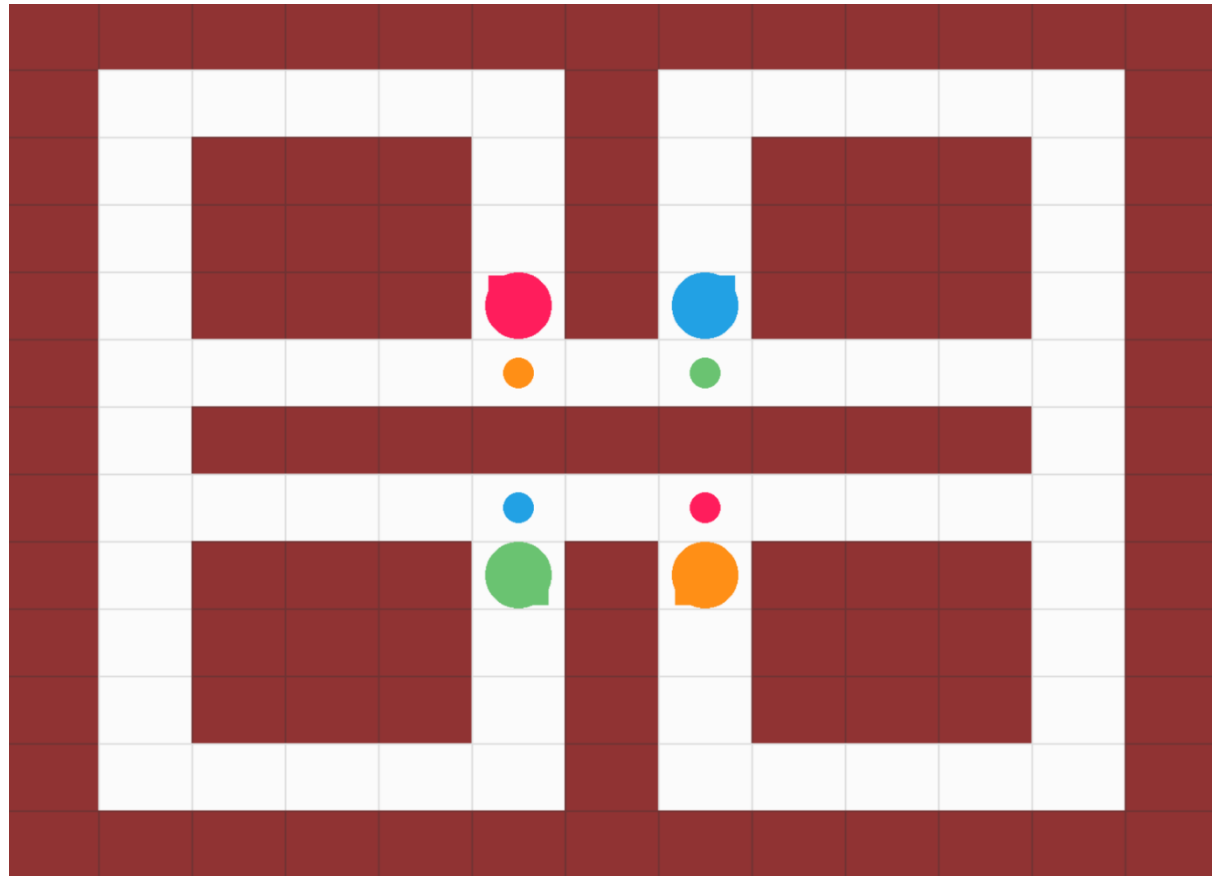
- Sie möchten Urlaub machen? → Stellen Sie einen Antrag!
 - Sie möchten versetzt werden? → Stellen Sie einen Antrag!
 - Sie möchten Ihren PKW anmelden? → Stellen Sie einen Antrag!

 - Wie finde ich mich in der Behörde zurecht?
 - Wo ist mein zuständiger Sachbearbeiter?
 - Welche Formulare brauche ich?
 - Habe ich Sie in der richtigen Reihenfolge eingereicht?
- ➔ Ein programmierter BOT soll mir die Aufgabe abnehmen!!!

GAME – BESTANDTEILE

Java Bots navigieren durch die Behörde

BEHÖRDENLABYRINTH

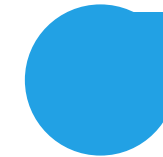


■ Wand □ Freies Feld ● Sachbearbeiter (SB)

JAVA-BOTS



SPIELER 1



SPIELER 2



SPIELER 3



SPIELER 4

PLAY – VERLAUF

Rundenbasierte Ausführung von Aktionen

Initiale Startinformationen an die Bots

Rundenbasierte Ausführung

- Statusinformationen an jeden Bot
- Aktionsausgabe / Befehl jedes Bots
- Ausführung der Aktionen in Startreihenfolge der Bots



Spielende

- Zielerreichung durch (min.) einen Bot oder
- Maximale Anzahl an Zügen gespielt

PROTOCOL – KOMMUNIKATION

Allgemeines Protokoll für die Initialisierung und je Runde

INIT

- `<maze info>`
 - `"<sizeX> <sizeY> <level>"`
- `<player info>`
 - `"<id> <startX> <startY>"`

Kommunikation per
Standardeingabe (**System.in**) und
Standardausgabe (**System.out**)

TURN

Input

- `<lastActionResult>`
- `<cell status (current)>`
- `<cell status (north)>`
- `<cell status (east)>`
- `<cell status (south)>`
- `<cell status (west)>`

Output

- `<action>`

DEVELOPE – BOT-CODE

Erstellen eines BOTS als ausführbares JAVA-Programm

ENTWICKLUNGSSCHRITTE

Programm mit **main**-Methode

Einlesen der INIT-Daten
Scanner / **System.in**

Wiederholtes Einlesen der TURN-Daten
(je Zug)

Wiederholte Ausgabe der Aktion
(je Zug)

Export als ausführbare JAR-Datei

BEISPIEL-BOT

MinimalBot.zip

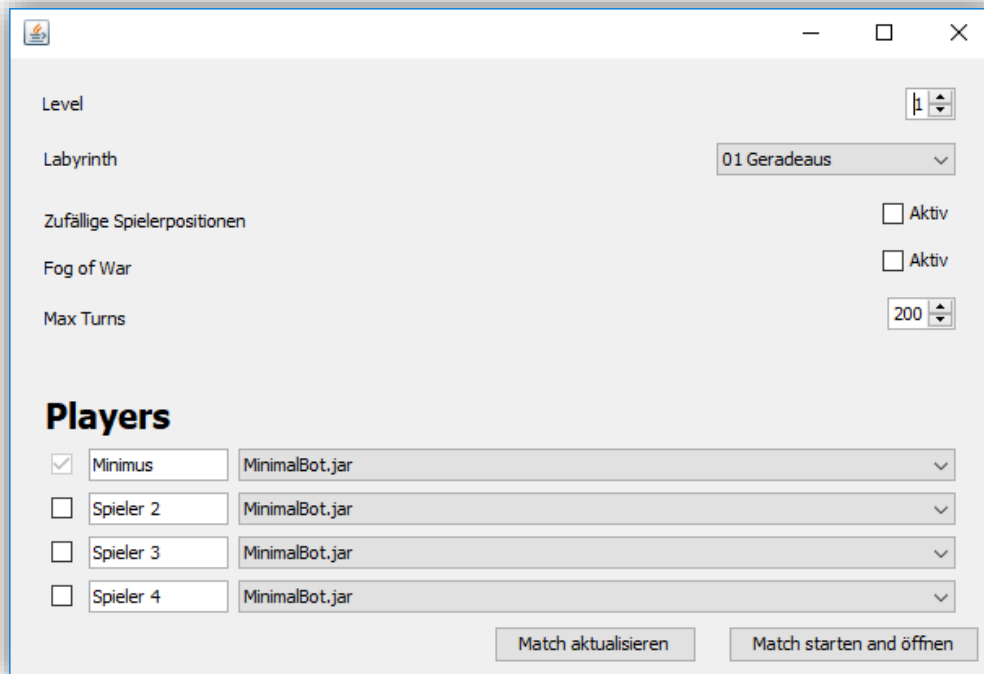
```
public static void main(String[] args) {  
    Scanner input = new Scanner(System.in);  
    // INIT  
    int sizeX = input.nextInt();  
    int sizeY = input.nextInt();  
    int level = input.nextInt();  
    input.nextLine();  
    // ...  
    // TURN (Wiederholung je Runde notwendig)  
    String lastActionResult = input.nextLine();  
    // ...  
    // Rundenaktion ausgeben  
    System.out.println("go west");  
    // ...  
}
```

SETUP – BOT-AUSFÜHRUNG

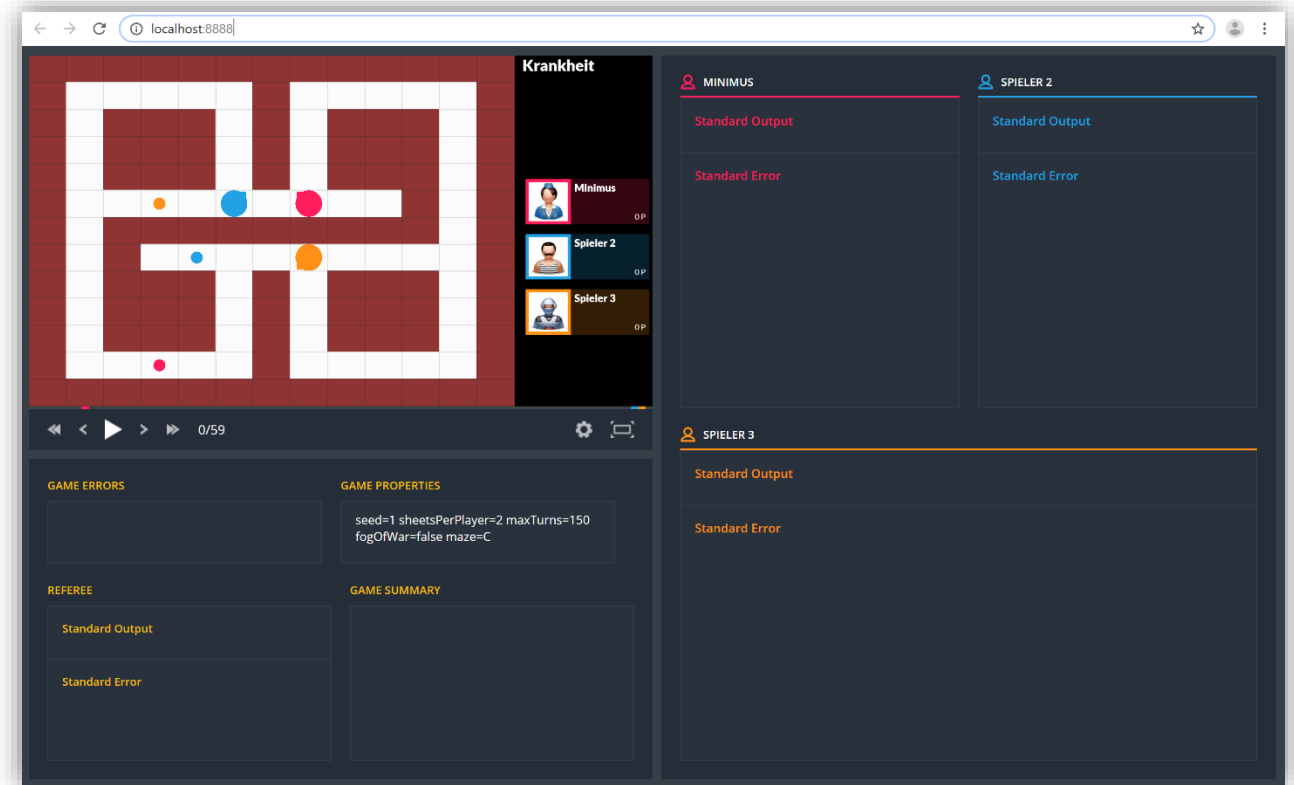
VITMaze Runner und Viewer

FHMAZE RUNNER

- Laufzeitumgebung für FHMaze (**FHMaze.zip**)
- Bot-JAR-Datei in Players-Ordner ablegen
- Spiel / Match konfigurieren (**FHMaze.jar**)
- Match starten (und öffnen)



FHMAZE VIEWER (IM BROWSER)



Spielanzeige im Browser (<http://localhost:8888>)