Hochschule des Bundes
für öffentliche
Verwaltung

Verwaltungsinformatik

JAVA CODING CHALLENGE

VITMaze

Digitalisierung im Verwaltungslabyrinth

Prof. Dr. Stalljohann, Dr. Merten
M08 – H2 2020

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?

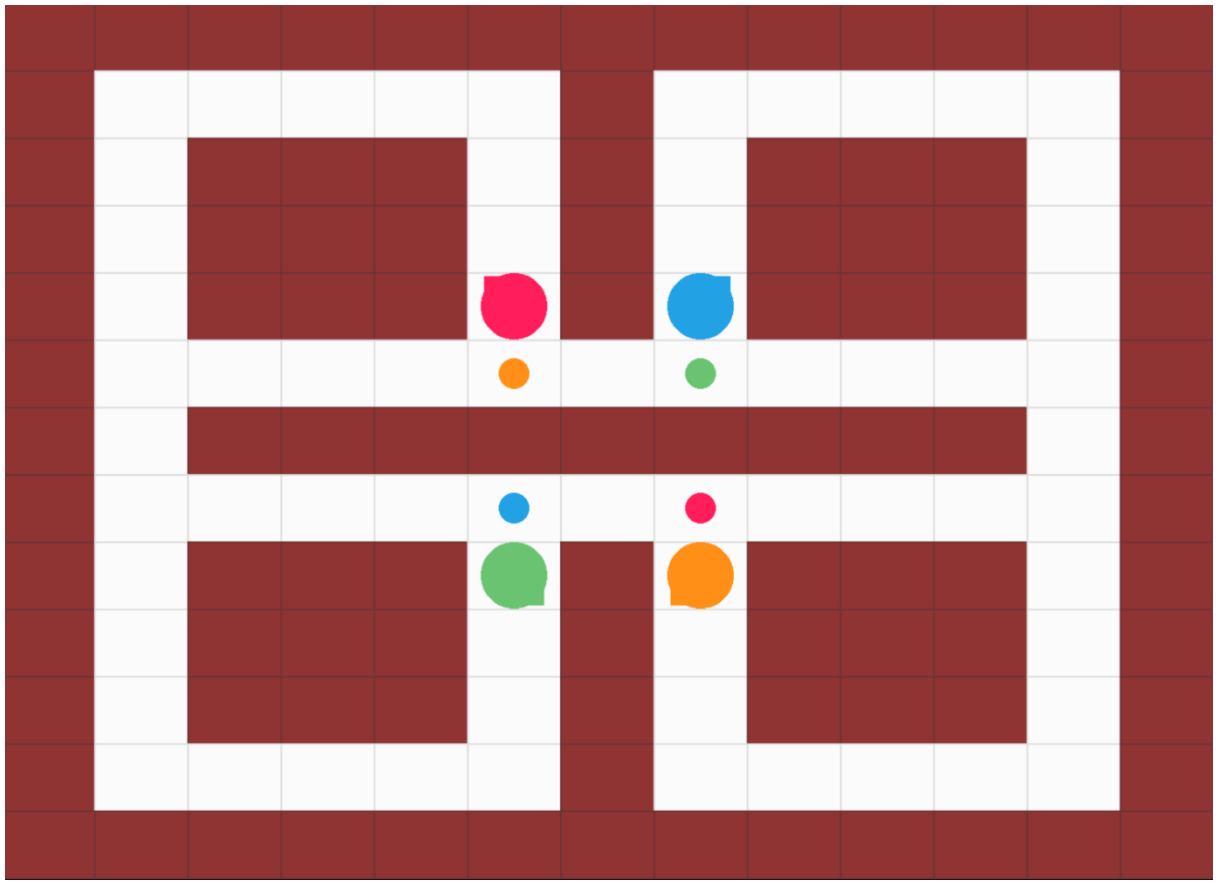
➔Ein programmierter BOT soll mir die Aufgabe abnehmen!!!

➔Mit künstlicher Intelligenz digitalisieren wir die Verwaltung!!! ;-)

GAME – BESTANDTEILE

Java Bots navigieren mit KI durch die Behörde

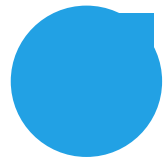
BEHÖRDENLABYRINTHE



Wand Freies Feld Sachbearbeiter (SB)
(hier für Spieler 1)

JAVA-BOTS

SPIELER 1



SPIELER 2

SPIELER 3

SPIELER 4

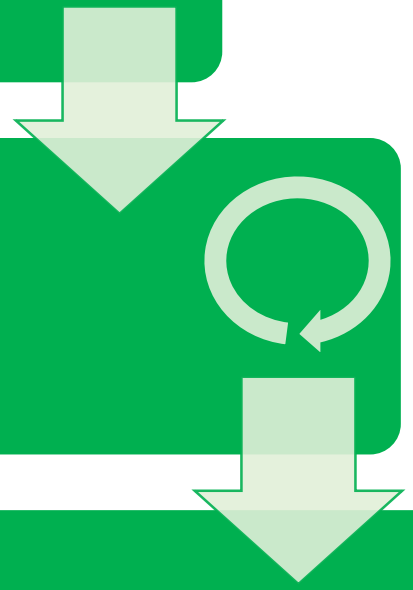
PLAY – VERLAUF

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`

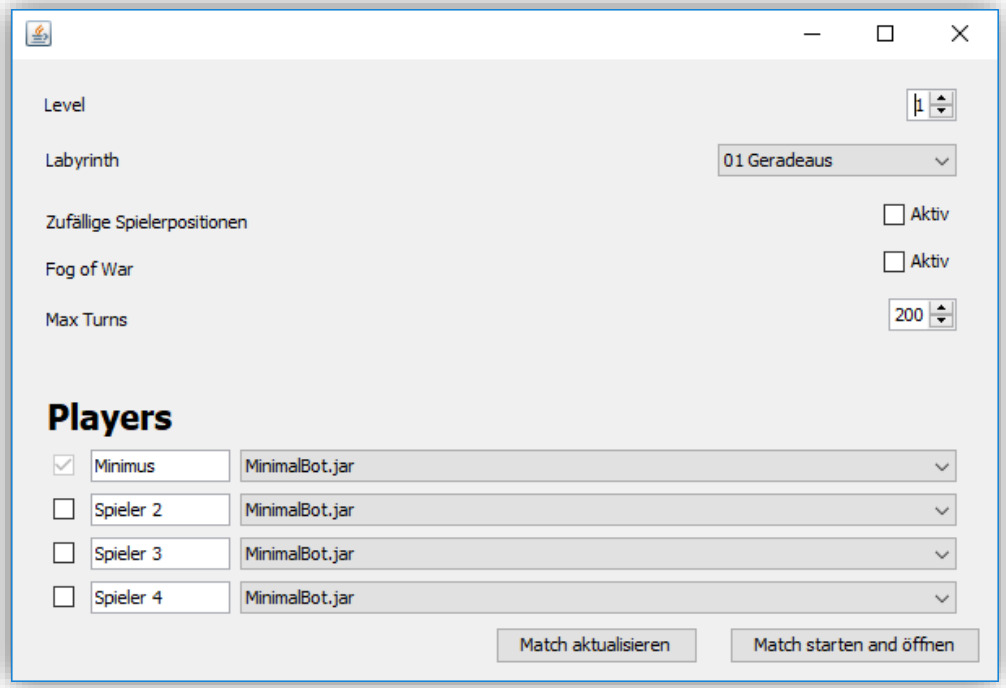
```
public static void main(String[] args) {
    Scanner input = new Scanner(System.in);
    // INIT
    int sizeX = input.nextInt();
    int sizeY = input.nextInt();
    int level = input.nextInt();
    input.nextLine();
    // …
    // TURN (Wiederholung je Runde notwendig)
    String lastActionsResult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

SETUP – BOT-AUSFÜHRUNG

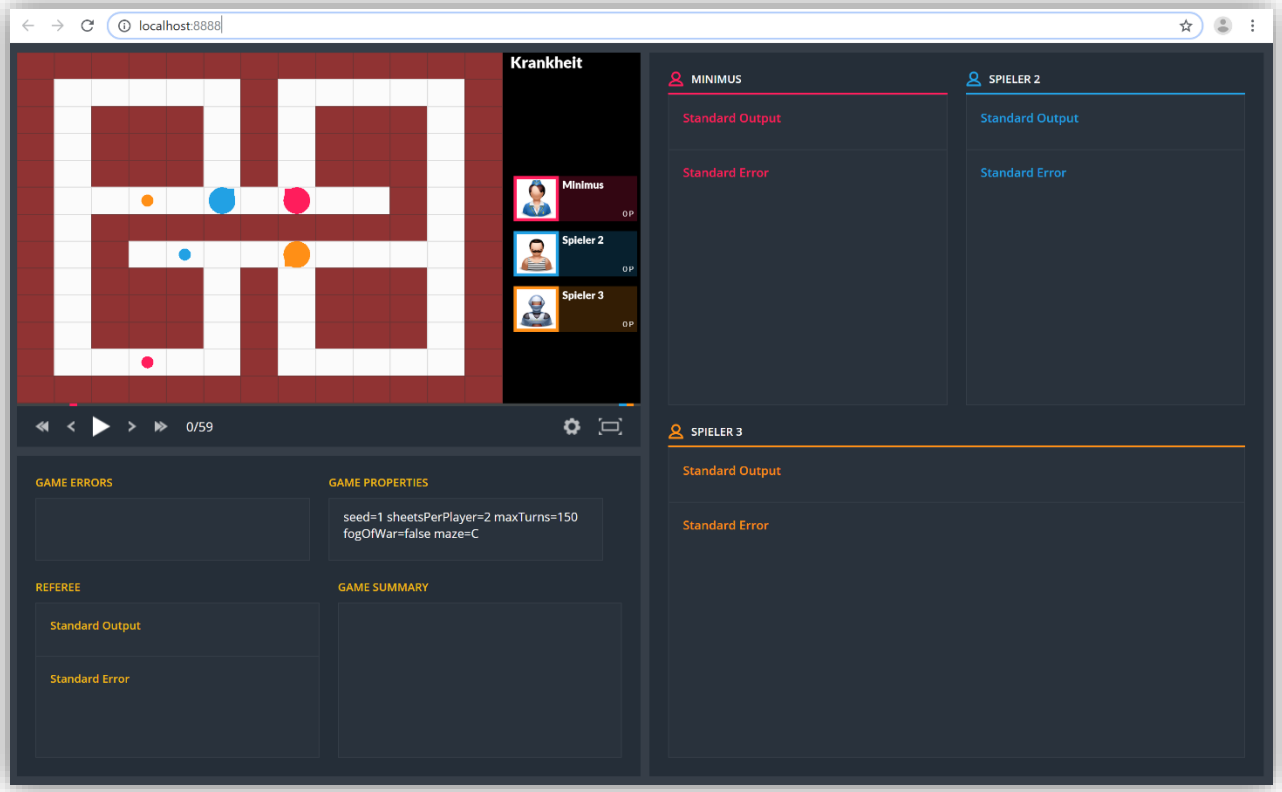
VITMaze Runner und Viewer

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

VITMAZE RUNNER



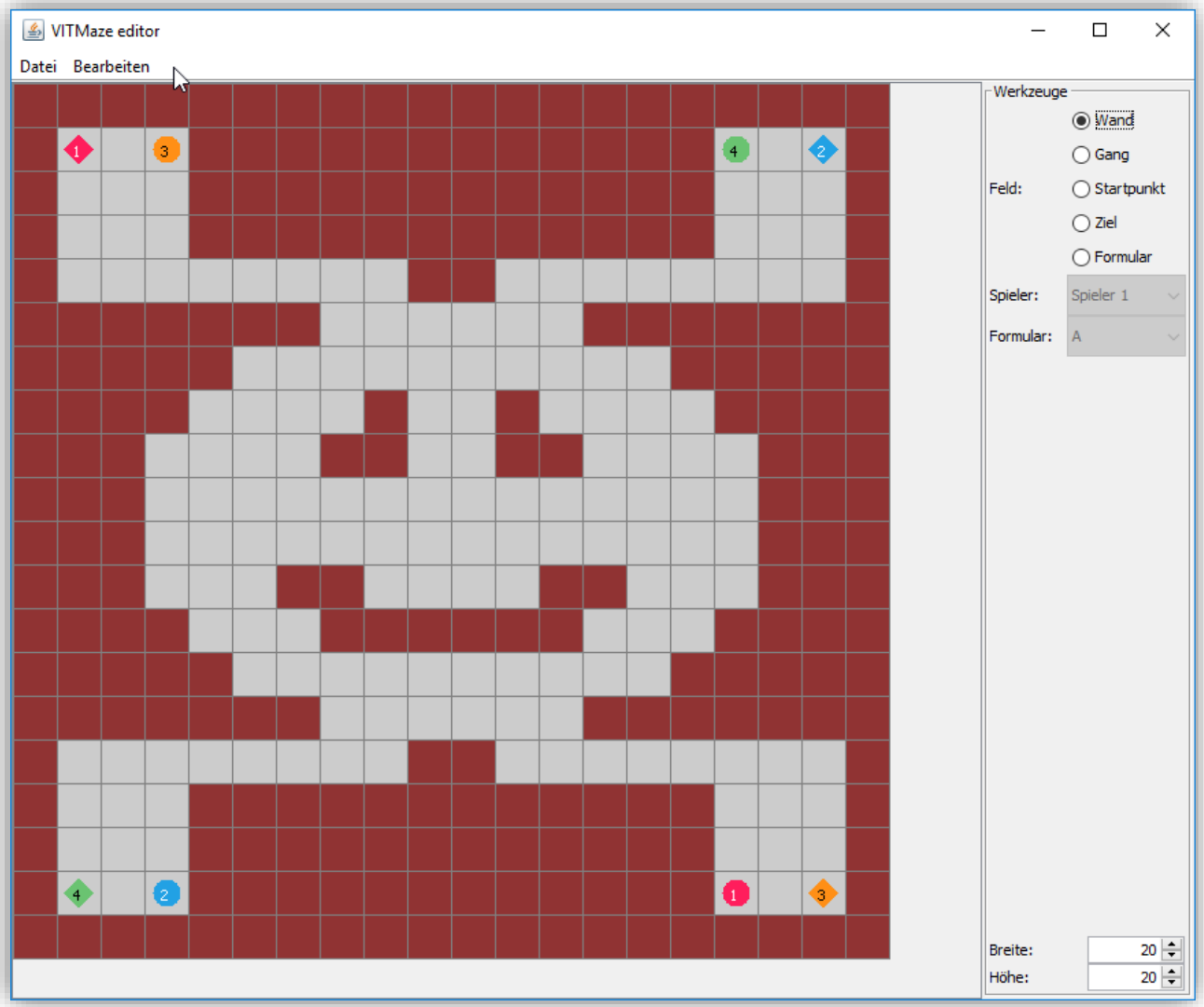
VITMAZE VIEWER (IM BROWSER)



- Spielanzeige im Browser (**`http://localhost:8888`**)

EDITOR - LABYRINTHE

Für Ihre eigenen Experimente



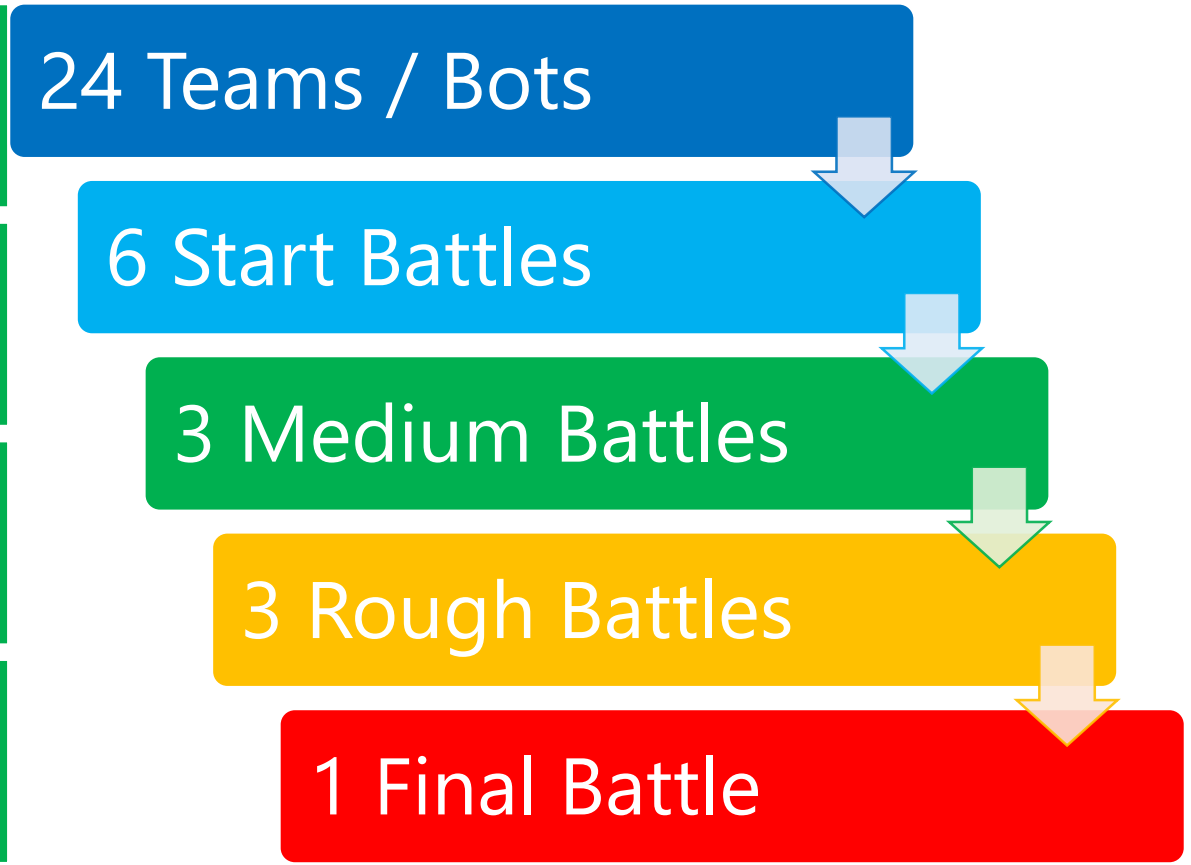
CHALLENGE – DAS TURNIER

21.07.2020 – 14:00 Uhr

TEAMS

- 2er Kleingruppen (selbstbestimmt)
- 4er Gruppen (zufällig)
- Gemeinsame Bot-Entwicklung
- Abgabe bis zum 21.07. 10:00 Uhr (ILIAS)

TURNIERMODUS



BATTLE - TURNIERSTUFEN

Start	Medium	Rough	Final
• **A - F** • Je 4 Bots (zufällig)	• **G** `[A1,D1,B2,E2]` • **H** `[B1,E1,C2,F2]` • **I** `[C1,F1,D2,A2]`	• **J** `[G1,H2,O1,O2]` • **K** `[H1,I2,O1,O2]` • **L** `[I1,G2,O1,O2]`	• **`[J1,K1,L1,O3]`**
TOP 2 je Battle	TOP 2 je Battle	TOP 1 je Battle	
• Bekannte Labyrinthe • Level 2	• Unbekannte Labyrinthe • Level 3	• Unbekannte Labyrinthe • Level 4 • Opponent-Bots O1 und O2	• Unbekannte Labyrinthe • Level 5 • Opponent-Bot O3

3 Matches je Battle mit ggf. verschiedenen Labyrinthen und Positionen
➔Platzierung nach Gesamtpunkten, Losentscheid bei Gleichstand

WIN – DER POKAL

RATING - BEWERTUNG

Abgabe (je Team)

- Bytecode (ausführbar)
- Quellcode (kommentiert)
- JavaDoc

Bewertung

Funktionalität

- Ausführbarkeit
- Regelkonformität
- Logik / Effektivität

Code Qualität

- Lesbarkeit / Wartbarkeit
- Strukturierung / Wiederverwendung
- Fehlertoleranz
- Effizienz

10% der Modulnote

Die Spielregeln

Digitalisierung in Leveln

LEVEL

Level 1	Level 2	Level 3	Level 4	Level 5
• Finden • jetzt	• Sammeln • 01.07. • 8:30 Uhr	• Unterhalten • 07.07. • 8:30 Uhr	• Kicken • 08.07. • 8:30 Uhr	• Verdecken • 15.07. • 8:30 Uhr

LEVEL 1 – FINDEN

Auf der Suche nach dem richtigen Sachbearbeiter

Finden Sie den für Sie zuständigen Sachbearbeiter und stellen Sie den Antrag!

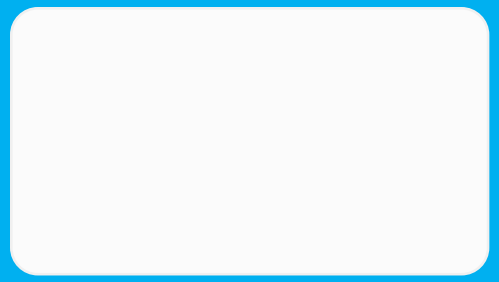
- Navigieren durch das Labyrinth
- Finden des eigenen Sachbearbeiters
- **100** Punkte bei Abschluss
- Kein Abschluss nach maximaler Rundenzahl
  - Unentschieden
  - **20** Punkte für „Last Player Standing“ bei mehr als einem Player

CELL STATUS – LEVEL 1

Status des aktuellen Feldes und der Nachbarfelder

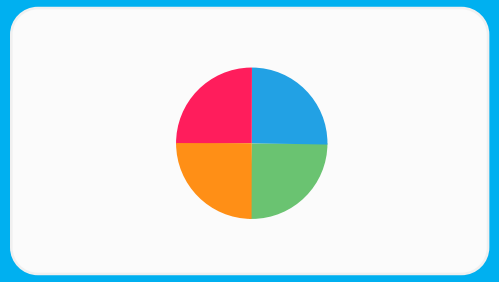
WALL

- **Wand**



FLOOR

- **Freies Feld / Gang**



FINISH <playerId> 0

- **Sachbearbeiter (SB) eines Spielers**

ACTION – LEVEL 1

ACTIONS		ACTION RESULTS
<erste Runde>		**OK**
go <direction>	erfolgreich	**OK <DIRECTION>**
north east	blockiert	**NOK BLOCKED**
south west		
position		**OK <X> <Y>**
finish	ihr SB	**OK**
	anderer SB	**NOK NOTYOURS**
	kein SB	**NOK EMPTY**
<sonstiges>		**NOK NOTSUPPORTED**

- Laufzeitfehler im Bot
- Antwortdauer
  - erste Runde > 1000ms
  - andere Runde > 50ms

➔ Deaktivierung des Bots