



JAVA CODING CHALLENGE

VITMaze

Digitalisierung im Verwaltungslabyrinth

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

STORY – MOTIVATION

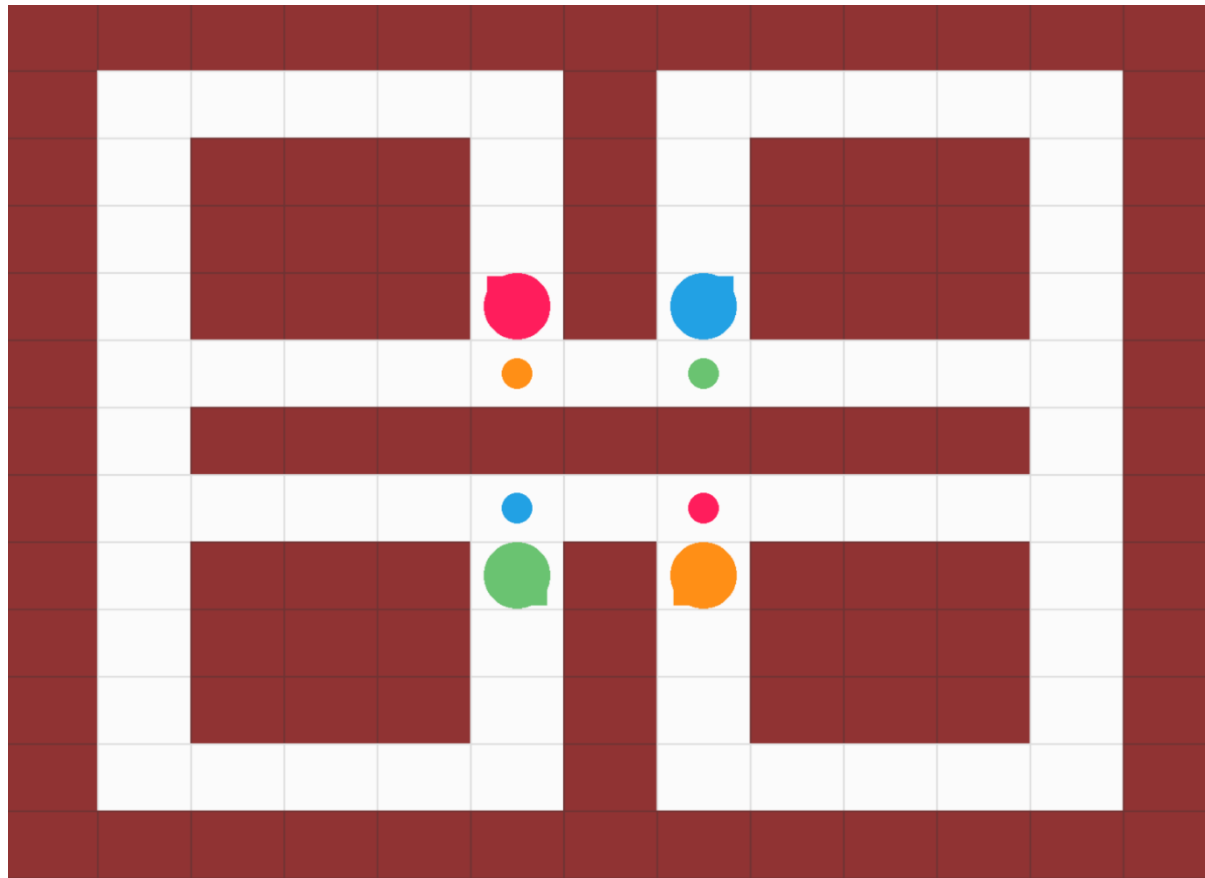
Wo bin ich, was mache ich und warum?

- 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

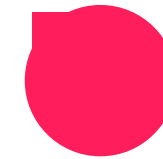
Java Bots navigieren mit KI durch die Behörde

BEHÖRDENLABYRINTHE

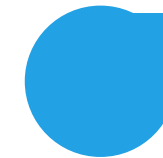


 Wand  Freies Feld  Sachbearbeiter (SB)
(hier für Spieler 1)

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

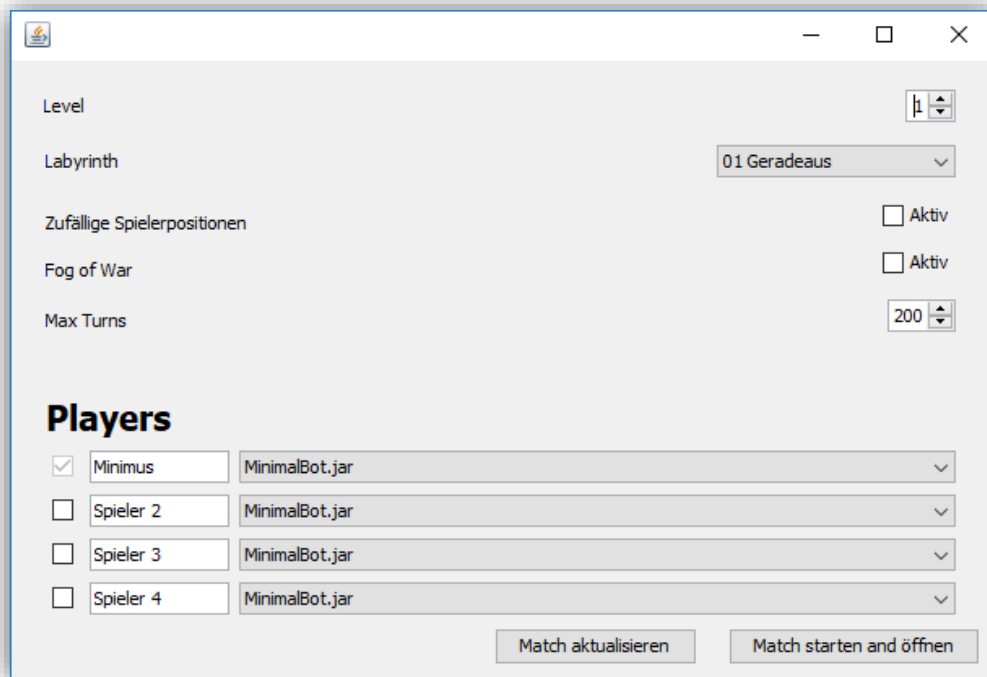
```
public static void main(String[] args) {  
    Scanner input = new Scanner(System.in);  
    // INIT  
    int sizeX = input.nextInt();  
    int sizeY = input.nextInt();  
    int level = input.nextInt();  
    input.nextLine();  
    // ...  
    // TURN (Wiederholung je Runde notwendig)  
   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

SETUP – BOT-AUSFÜHRUNG

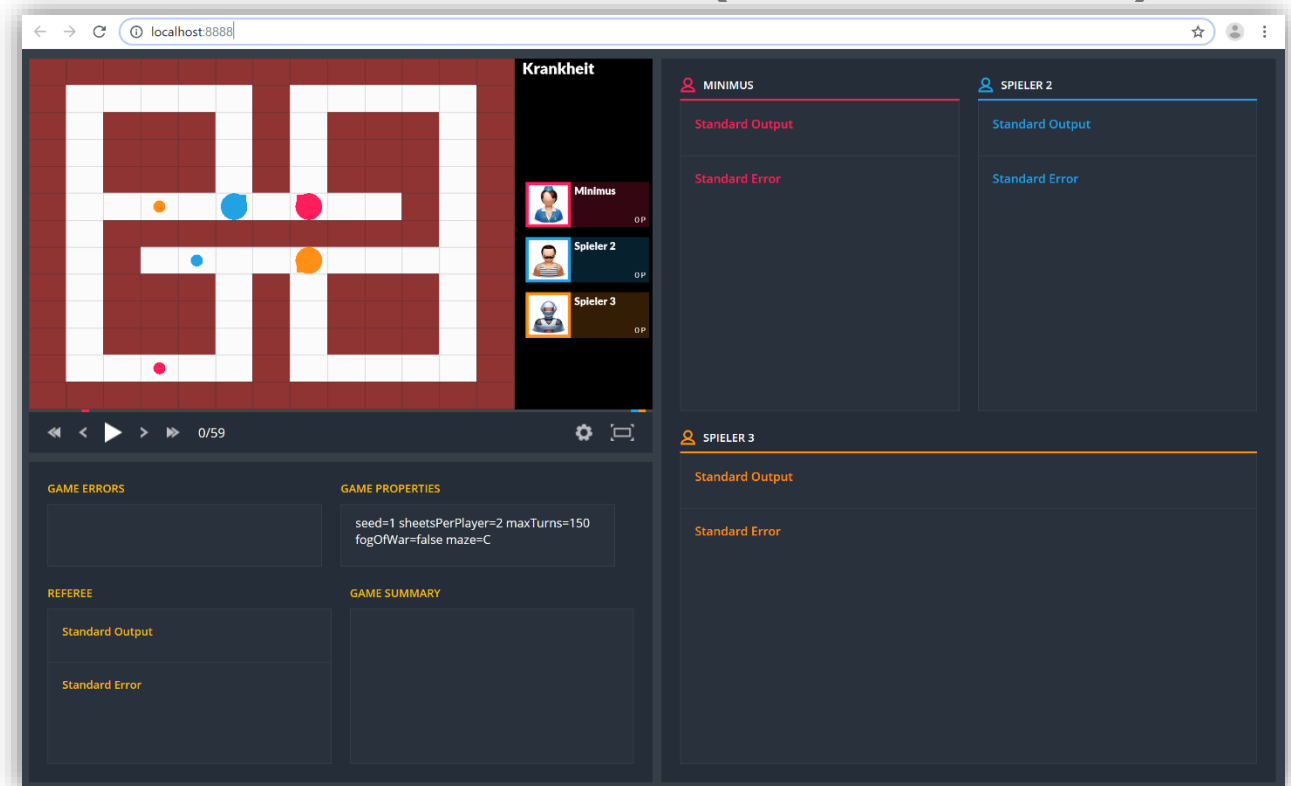
VITMaze Runner und Viewer

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

VITMAZE RUNNER



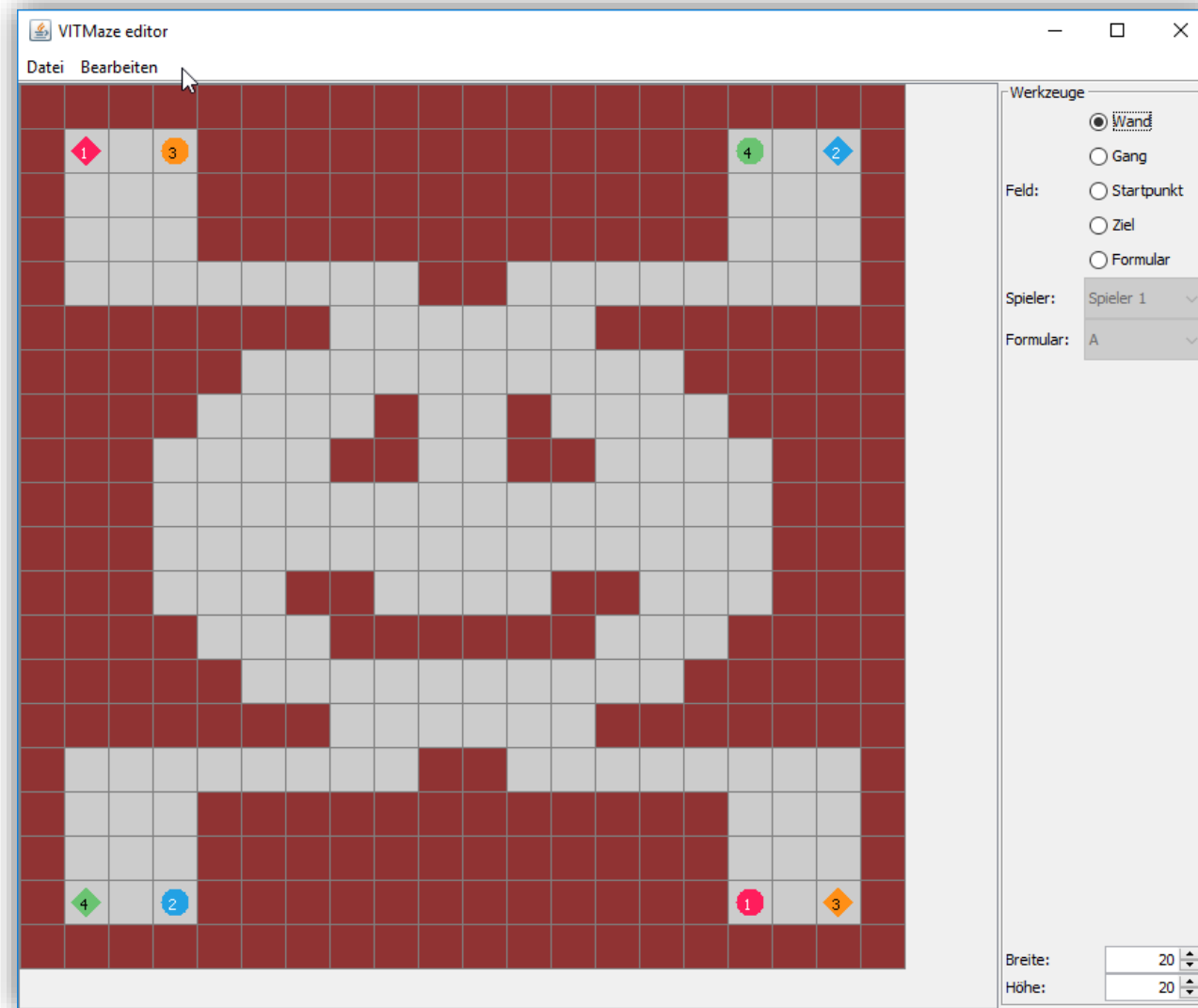
VITMAZE VIEWER (IM BROWSER)



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

EDITOR - LABYRINTHE

Für Ihre eigenen Experimente



CHALLENGE – DAS TURNIER

29.07.2019 – 14:00 Uhr

TEAMS

2er Kleingruppen (selbstbestimmt)

4er Gruppen (zufällig)

Gemeinsame Bot-Entwicklung

Abgabe bis zum 29.07. 8:00 Uhr
(ILIAS)

TURNIERMODUS

24 Teams / Bots

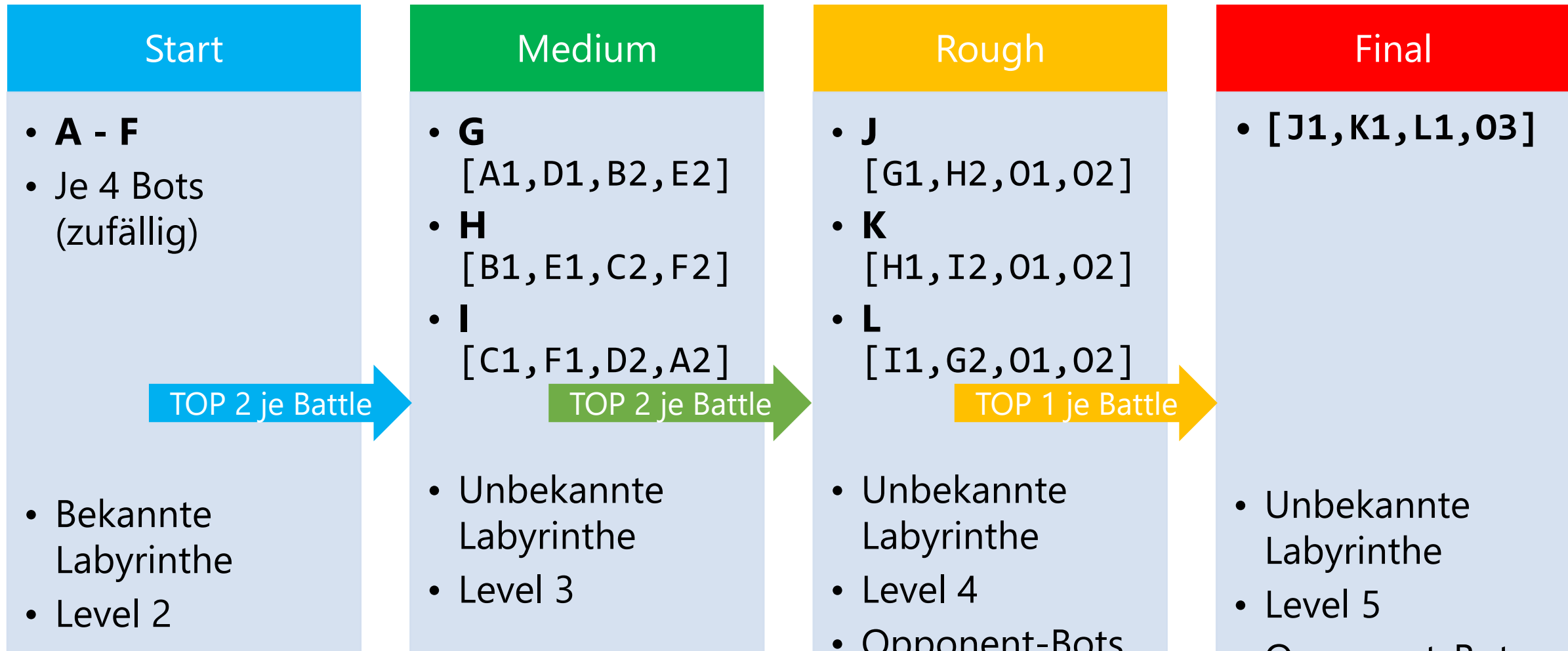
6 Start Battles

3 Medium Battles

3 Rough Battles

1 Final Battle

BATTLE - TURNIERSTUFEN



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

- Finden
- jetzt

Level 2

- Sammeln
- 01.07.
- 8:30 Uhr

Level 3

- Unterhalten
- 07.07.
- 8:30 Uhr

Level 4

- Kicken
- 08.07.
- 8:30 Uhr

Level 5

- 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

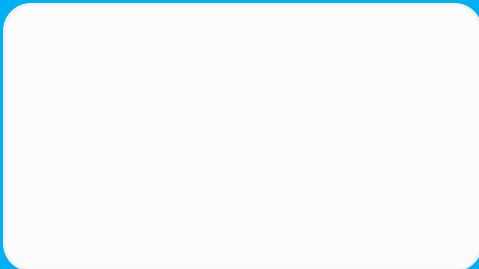
CELL STATUS – LEVEL 1

Status des aktuellen Feldes und der Nachbarfelder



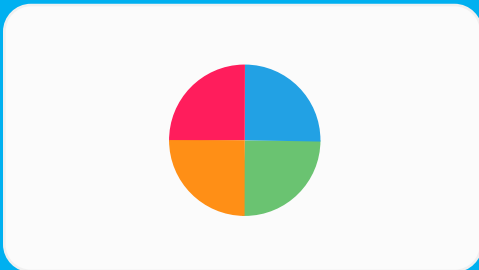
WALL

- Wand



FLOOR

- Freies Feld / Gang



FINISH `<playerId>` 0

- Sachbearbeiter (SB) eines Spielers

ACTION – LEVEL 1

ACTIONS

<erste Runde>

go <direction>

north east

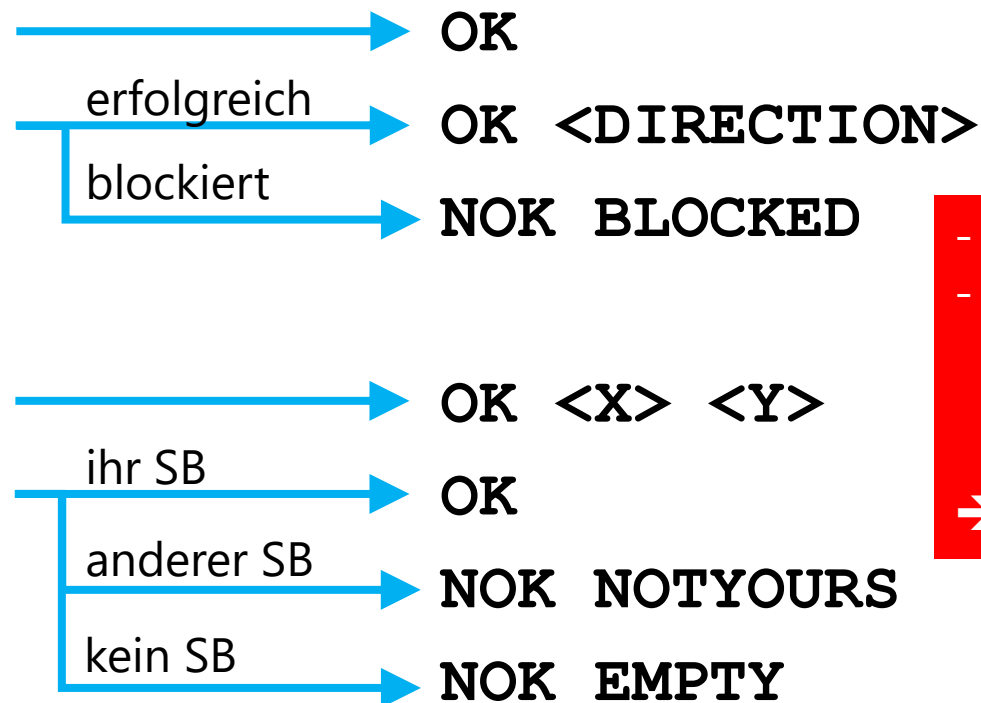
south west

position

finish

<sonstiges>

ACTION RESULTS



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