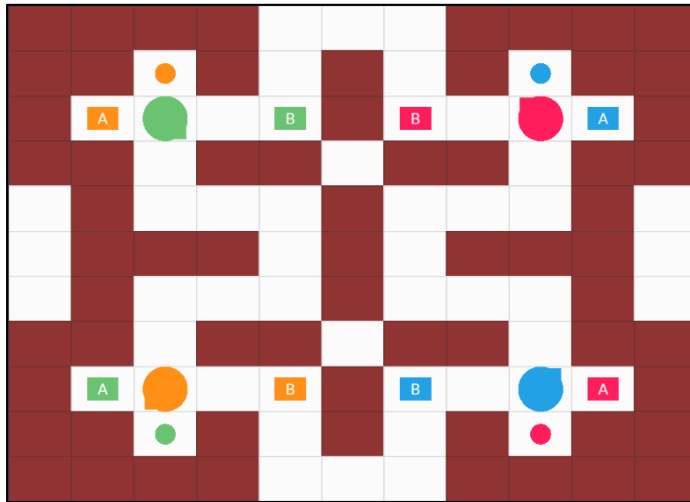


# Praktikum 3: Unterhalten

Stand: 11.11.2025



## Kontext

- Sie erweitern ein Java-Programm, das als Spieler an Labyrinthläufen teilnehmen kann.
- Das Protokoll und die Spielregeln erweitern sich schrittweise von Praktikum zu Praktikum.
- In Level 3 können sich Bots sehen und auf Feldern begegnen. Bei Begegnungen blockieren Sie sich kurz gegenseitig.

## Ziele

- Laufähiges Programm ohne randomisierte Anteile für die Laufzeitumgebung von FHMaze, dass die Level 1 und 3 in den Labyrinthen 01 bis 10 spielen kann.
- Weitere Entkopplung der konkret Verwendeten Klassen und Realisierung von Erweiterungsmechanismen für die neu hinzugekommenen Regelbestandteile anhand der SOLID-Prinzipien und unter Einsatz geeigneter Entwurfsmuster.
- Erhöhung der isolierten Testbarkeit (JUnit-Tests).
- Dokumentation des Entwurfs als UML-Klassendiagramm.

## Randbedingungen

- Bearbeiten Sie die Aufgaben in den zugeteilten **4er/3er Teams**.
- Jedes Teammitglied muss den Code sowie den Entwurf vollständig erklären und Fragen dazu beantworten können. Die Vorlage rein fremd erstellter oder KI-generierter Lösungen genügt nicht für das Testat.
- Die Verwendung von KI-Tools (ChatGPT, Copilot etc.) darf nur zur Recherche, Erläuterung und Klärung von Fragen sowie die Generierung von funktionalen Code-Fragmenten genutzt werden, sollte aber nur unterstützend wirken. Die Generierung von Strukturen des Entwurfs oder Berücksichtigung von Prinzipien oder Mustern ist nicht gestattet.
- Erfüllt die Abgabe nicht sämtlichen Anforderungen der Pflichtaufgaben oder kann ein Teammitglied nicht sämtlichen Code erläutern, gilt das Testat als nicht bestanden.

## Hinweise

- Diskutieren Sie Fragen zu den Aufgaben auch gerne im Mattermost-Chat.
- Laufzeitfehler in der Ausführung des Bots führen zu seiner Deaktivierung.
- Nach Überschreiten einer tolerierten Antwortdauer wird der Bot ebenfalls deaktiviert.
  - erste Runde inkl. Initialisierung: max. 1000 ms
  - weitere Runden: max. 50 ms

## LEVEL 3 - Unterhalten

Plaudern ist ein Zeitvertreib!

### Task

#### Berücksichtigen Sie die Kollision mit anderen Bots!

- Die Sichtbarkeit von anderen Bots in direkter Linie wird ergänzt.
- Sind zwei oder mehr Bots nach Ausführung jeder Aktion einer Runde auf dem selben Feld, beginnt ein „Gespräch“. Vorbeigehen durch Platzwechsel in der gleichen Runde ist möglich.
- In der direkt nachfolgenden Runde ist die Aktion dieser Bots unwirksam.
- Sie erhalten je **-5 Punkte** pro „Gespräch“.
- Die weiteren Regeln und Punkte aus Level 1 und 2 bleiben erhalten.

### Cell Status

In diesem Level kommen keine neuen Cell Status hinzu, das Protokoll zu jedem Status wird jedoch erweitert.

- Feld mit Gegner:  
**<status> !**
- Nachbarfeld (north,east,south,west) mit „sichtbarem“ Gegner in diese Richtung:  
**<status> !<distance>**

Beispiele:

**FLOOR!** Gang-Feld mit Gegner

**FORM 1 1 !2** Feld mit Formular A von Spieler 1 und Gegner(n) zwei Felder weiter als dieses Nachbarfeld in gleicher Richtung

### Actions

Erweiterungen und Änderungen zu Level 1 und 2 bzgl. möglicher **<action>**, die der Bot als Ergebnis eines Spielzugs ausgeben kann sowie zugehörige Ergebnisse **<lastActionResult>** der Folgerunde:

ACTIONS

ACTION RESULTS

**<action>**

„Zusammenstoß“ mit anderem  
Bot in vorheriger Runde

**NOK TALKING**

## Ablauf

Ihr Praktikumstermin setzt sich aus den folgenden Bestandteilen und Tätigkeiten zusammen:

### 1. Vorbereitung (vor Praktikumsbeginn)

- Arbeiten Sie sich in das Themengebiet der Entwurfsmuster ein.
- Bereiten Sie die Anlage von UnitTests in Ihrer Umgebung vor.  
Hinweise: [Get started with JUnit | IntelliJ IDEA Documentation](#)
- Beschäftigen Sie sich mit den neuen Regeln, Zielen und Randbedingungen.

### 2. Erläuterung der Aufgabenstellung (ca. 10 Minuten)

- Kurzvorstellung der Praktikumsaufgabe durch einen Tutor
- Diskussion und Rückfragemöglichkeiten der Teilnehmer

### 3. Durchführung (ca. 3 Stunden)

- Überarbeitung (Refactoring) des Entwurfs sowie des zugehörigen UML-Klassendiagramms gemäß der folgenden Kernaspekte:
  - Umsetzung der Regeln für Level 3 (Level 1 und 2 sollen spielbar bleiben). Gegner sollen explizit berücksichtigt werden.
  - Weitere Reduktion der Kopplung zwischen (konkreten) Klassen unter Berücksichtigung der SOLID-Prinzipien und geeigneter Entwurfsmuster.
  - Erhöhung der isolierten Testbarkeit einzelner Klassen und Methoden.
- Aktualisierung der Implementierung der Software auf Basis des Entwurfs.
- Erstellung geeigneter JUnit-Tests, welche die isolierte Testbarkeit der wesentlichen Bestandteile zur Initialisierung, Aktualisierung und Entscheidungsfindung demonstrieren.
- Diskussion von Schwierigkeiten, Unklarheiten und Alternativen im Team
- Rückfragen und Feedback während der Bearbeitung

### 4. Abtestat (ca. 60 Minuten)

- Jedes Team hat ca. 15 Minuten Zeit für das Testat
- Voraussetzungen
  - Der Bot beendet mindestens die Labyrinth 01 bis 10 erfolgreich für Level 1, 2 und 3 und berücksichtigt Gegner aktiv.
  - Der Code ist in der GitLab-Gruppe des Teams eingecheckt
  - Es liegt ein vollständiges UML-Klassendiagramm vor.
- Das Team stellt den Entwurf mit Bezug zu Modifikation und Restrukturierungen kurz vor und erläutert Rückfragen
- Jedes Teammitglied leistet einen Redebeitrag, insbesondere in Bezug auf Entwurfsmuster, SOLID-Prinzipien, isolierte Testbarkeit sowie Entkopplung der Klassen.
- Ggf. Nacharbeit und Aktualisierung des Codes und des Diagramms gemäß den Absprachen im Testat