

## Das Java-Projekt für das KI-Praktikum

Das Java-Projekt PacmanAgent besteht aus einem Server (Server.jar) zur Simulation der Agentenumgebung sowie einem Client (MyAgent.jar), letztere sollen im Verlauf des Praktikums um verschiedene Funktionalitäten erweitert werden.

## Projekt-Konfiguration unter Eclipse

Folgende Anleitung beschreibt exemplarisch, wie ihr unser Projekt in die Java Entwicklungsumgebung Eclipse einbinden könnt. Ihr könnt selbstverständlich einen Java Editor/IDE eurer Wahl verwenden.

Ladet euch das Projekt-Zip herunter.

Führt `File → Import... → General → Existing Projects into Workspace` aus und wählt unter `Select archive file` das Zip-Archiv aus. (Dadurch wird das Zip-Archiv in eurer Eclipse-Arbeitsverzeichnis entpackt.) Danach sollte sich das Projekt in eurem Eclipse Package Explorer befinden.

### Hinweis:

Einen Überblick (über die einzelnen Aufgaben der Praktikumsblätter) bekommt ihr, wenn ihr euch unter `Windows → Show View → Tasks` die `TODO` Markierungen im Code anschaut.

### Libraries überprüfen und ggf. einstellen:

Das Eclipse-Projekt ist i.d.R. bereits vorkonfiguriert. Bitte überprüft folgende Einstellungen und ändert diese gegebenenfalls:

Um den Client für Pac-Man zu konfigurieren, führt folgende Aktion auf dem Projekt `PacmanAgent` aus (wenn nicht schon eingestellt):

`Project → Properties → Java Build Path → Libraries → AddJARS... → PROJEKTNAME → Server.jar`

### Den Server starten:

Um euren Agenten in der Welt spielen lassen zu können, müsst ihr zunächst den Server starten. Um den Server zu starten, führt folgende Schritte durch:

Selektiert die `Server.jar` in eurem `Package Explorer` und wählt im Menü

`Run -> Run as -> Java Application`

Weitere Einstellungen könnt ihr unter der grafischen Benutzeroberfläche des Servers einstellen.

### Den Client starten:

Um den Client nutzen zu können, führt folgende Schritte durch:

Als erstes startet die `Server.jar` (s.o.).

Führt unter Eclipse im Package Explorer die Datei `de.fh.MyAgent.java` aus.

## Projekt-Konfiguration unter IntelliJ

Ladet euch das Projekt-Zip herunter und entpackt es.

In IntelliJ öffnet den entpackten Ordner unter `File → Open... → Im Dialog zum Ordner navigieren` (Der Ordner sollte im Auswahldialog mit einem schwarz-weißen Kästchen versehen sein).

### Hinweis:

Einen Überblick (über die einzelnen Aufgaben der Praktikumsblätter) bekommt ihr, wenn ihr in der Fußleiste der Konsole die Ansicht `TODO` auswählt (oder unter `View → Tool Windows → TODO`).

### Den Server starten:

Um euren Agenten in der Welt spielen lassen zu können, müsst ihr zunächst den Server starten. Um den Server zu starten, führt folgende Schritte durch:

Selektiert die `Server.jar` in eurem `Package Explorer` und wählt im Menü

`Run 'Server.jar'`

Ist in der Konsole folgender Fehler zu sehen:

```
The class javafx/application/Application could not be found.  
Please download OpenJFX from:  
https://openjfx.io  
and include it as a library.
```

Klicken Sie auf den Link und laden die für Ihr Betriebssystem passende JavaFX SDK herunter. Verschieben Sie diese in einen Ordner Ihrer Wahl, z.B. `C:\Program Files\Java\JFX` und entpacken Sie das zip-Archiv.

In IntelliJ auf der Rechten Seite der oberen Toolbar klicken Sie auf `Server.jar` (das Run-Konfigurationsfeld) → `Edit Configurations... → Server.jar → VM Options` und fügen Sie die folgenden Optionen hinzu (ersetzen Sie `Pfad\zur\JFX\lib` mit dem Pfad zum `\lib` Unterordner der JavaFX SDK, die Sie zuvor heruntergeladen haben):

```
--module-path "Pfad\zur\JFX\lib" --add-modules=javafx.controls,javafx.fxml
```

Der Server sollte jetzt einsatzbereit sein.

### Den Client starten:

Um den Client nutzen zu können, führt folgende Schritte durch:

Als erstes startet die `Server.jar` (s.o.).

Führt unter Eclipse im Package Explorer die Datei `de.fh.stud.MyAgent_P1.java` aus.

Die folgenden beiden Aktivitätsdiagramme (siehe Abbildung 1 & 2) zeigen das Zusammenspiel zwischen Client & Server:

#### Server:

- Nachdem der Server gestartet wird lauscht er auf sich verbindende Clients. Sobald ein Client verbunden ist (durch bspw. Starten des MyAgent), kann das Spiel gestartet werden.
- Der Server sendet initiale Startinformationen an alle teilnehmenden Clients und folgt anschließend einem Zyklus, bis das Spiel zu Ende ist.
- Eine Runde eines Zyklus läuft wie folgt ab: hintereinander sind alle teilnehmen Agenten an der Reihe. Der Server übermittelt jedem Agenten die Wahrnehmung seines Pac-Man, wartet auf die Rückmeldung des Agenten (in Form einer PacmanAction) und führt diese aus. Nachdem alle Agenten an der Reihe waren beginnt die nächste Simulationsrunde.

#### Client:

- Beim Starten des Clients versucht sich dieser mit einem Server zu verbinden. Konnte eine Verbindung hergestellt werden, beginnt der Zyklus des Clients.
- Über den Parameter percept der Methode action erhält der Agent in jeder Runde die aktuelle Wahrnehmung der Welt. Der Parameter actionEffect stellt die Rückmeldung des Servers auf die zuletzt ausgeführte Aktion dar.
- Der Agent teilt dem Server seine nächste Aktion durch das zurückgegebene PacmanAction Objekt der action-Methode mit.

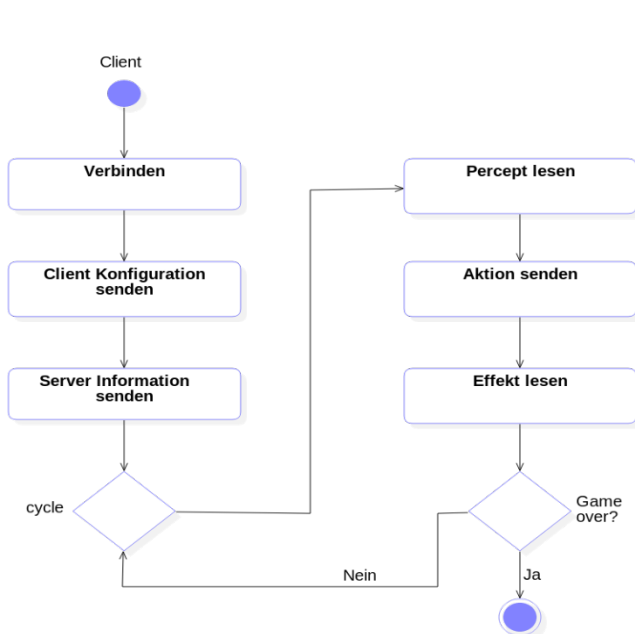


Abbildung 1: Der Simulationszyklus des Clients.

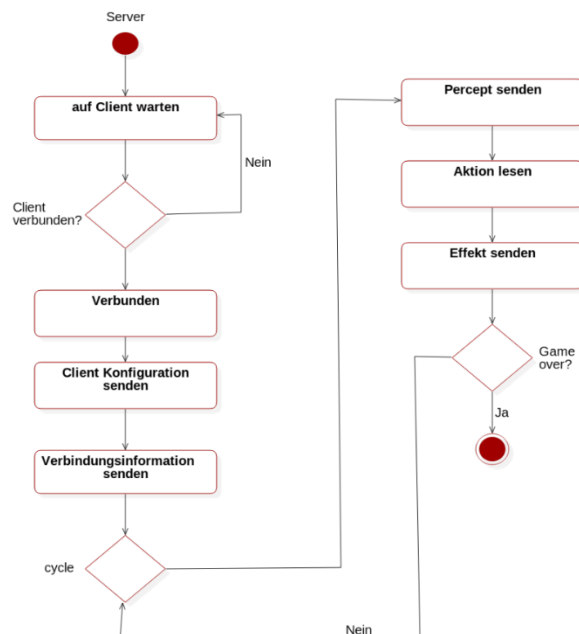


Abbildung 2: Der Simulationszyklus des Servers.

## Tipps & Tricks zur Bedienung des Servers

- Zum Editor des Servers gehören die verschiedenen Werkzeuge und die Modi der Werkzeuge. Über die Werkzeuge (Wall, Dot, Pac-Man etc.) lässt sich die Pac-Man-Welt im Editor verändern.
- Über die Tastatur lassen sich durch Drücken der **Zahlentasten** die verschiedenen Werkzeuge anwählen bzw. durch zusätzliches Drücken von **Strg** die verschiedenen Modi.
- Die Buttons Pause/Unpause und Restart können über die Tasten **P** und **R** direkt ausgeführt werden.
- Die meisten Funktionen/Menüs lassen sich auch über die **Pfeiltasten**, **Tab** und **Enter** bedienen.
- Durch einen **Rechtsklick** könnt ihr ein Kontextmenü öffnen, welche verschiedene Einstellungen der Spielfiguren (Pac-Man, Geister etc.) bietet.
- Über das Menü „**Start**“ -> „**Start multiple Runs**“ könnt ihr euren Agenten in vielen aufeinanderfolgenden Spielen in der gleichen Welt spielen lassen. Anschließend wird eine Statistik über die Spielergebnisse ausgegeben.