

## Praktikum 1: Einführung in den Server

### MyAgent & Pac-Man-Server

In diesem Praktikum arbeiten wir mit dem Pac-Man-Server und dem MyAgent-Client. Unser Pac-Man wird durch die Klasse *MyAgent* gesteuert. Die Klasse *MyAgent* stellt zusammen mit der *Pac-Man-Server.jar* eine Client-Server-Architektur dar.

Weiterführende Informationen zum Server, zur Bedienung, Architektur und Einrichtung findet Ihr in der *Server\_Readme*.

Weiterführende Informationen zur Aufgabe findet Ihr wie gewohnt im Hilfsblatt.

### Aufgabe P1

Der Pac-Man im Projekt läuft nur geradeaus nach Osten. Sobald er auf eine Wand trifft, läuft er permanent dagegen. Probiert es aus!

Kollidiert der Pac-Man mit einer Wand, soll er sich nach rechts (bzgl. seiner Laufrichtung), in diesem Fall nach Süden wenden, und dann wieder geradeaus weiter gehen ... geht es geradeaus nicht weiter, so läuft er als nächstes nach Westen ... usw.

#### *Nutzt dafür die besprochenen PacmanActionEffects:*

GAME\_INITIALIZED, GAME\_OVER, BUMPED\_INTO\_WALL, ATE\_DOT, MOVED\_ON\_EMPTY\_TILE und BUMPED\_INTO\_PACMAN, ATE\_POWERPILL, ATE\_GHOST, ATE\_GHOST\_AND\_DOT und ATE\_GHOST\_AND\_POWERPILL.

#### *und die möglichen PacmanActions:*

GO\_NORTH, GO\_EAST, GO\_SOUTH, GO\_WEST, WAIT, QUIT\_GAME

Implementiert diesen Pac-Man und testet ihn, bspw. in der Umgebung „PW\_easy\_01“.

### Zusatzaufgabe (für Interessierte):

Der oben beschriebene „Kreisläufer“ ist ein sehr einfacher Simple-Reflex-Agent und kann als solcher nur in sehr wenigen Welten alle Dots fressen. Verbessert den „Kreisläufer“, so dass er alle Dots in der oben beschriebenen Welt „PW\_easy\_01“ fressen kann. Versucht dafür nur wenige zusätzliche Regeln zu implementieren.