

## Praktikum 2: Knoten & Bäume

### Der Pac-Man und die Pac-Man Welt

Unser Pac-Man im KI-Praktikum spielt nicht das originale Pac-Man-Spiel. Stattdessen werden wir in einer Labyrinth-ähnlichen Umgebung mit verschiedenen Suchstrategien experimentieren.

Weiterführende Informationen findet ihr im Hilfsblatt.

In den folgenden Praktika werden die Suchen aus den Vorlesungen implementiert. Die Suchen bauen basierend auf der aktuellen Welt des Pac-Mans einen Baum auf, in dem jeder Knoten einen Weltzustand darstellt und die ausgeführten Aktionen die Verbindungen zwischen den Knoten.

### Aufgabe P2

In diesem Praktikum wird die Klasse *Knoten* entwickelt, die den Zustand der Welt des Pac-Mans beschreibt und zum Aufbauen eines Suchbaums genutzt wird. In den späteren Praktika werden diese Suchbäume entsprechend der verschiedenen Suchverfahren aufgebaut.

Folgende Frage stellt sich: Welche Informationen müssen die Knoten speichern, um die möglichen verschiedenen Weltzustände darstellen zu können und mit welchen Datentypen sollen diese gespeichert werden? Überlegt auch wie Knoten expandiert werden, d.h. wie neue Weltzustände aus einem Knoten generiert werden können (durch Aufruf der *expand*-Methode).

Die Belegung der einzelnen Felder der Welt des Pac-Man sind durch das enum *PacmanTileTypes* vorgegeben. *PacmanTileTypes* können sein:

WALL, DOT, EMPTY, PACMAN, GHOST, GHOST\_AND\_DOT, POWERPILL und GHOST\_AND\_POWERPILL

Der Pac-Man kann folgende Aktionen ausführen:

GO\_NORTH, GO\_EAST, GO\_SOUTH, GO\_WEST und WAIT

1. Erweitert die Klasse *Knoten* um entsprechende Attribute, Methoden und Konstruktoren.
2. Implementiert die *expand*-Methode. Diese soll die Kindknoten erzeugen und zurückgeben.
3. Expandiert anschließend basierend auf dem Anfangszustand (siehe Abbildung 1) die ersten 10 Knoten des Baums und gibt diese auf der Konsole aus. Dazu steht euch in der Klasse *BaumTest* eine main-Methode zur Verfügung.

Auf welches Problem stößt man hier beim Expandieren und wie kommt dieses zustande?

Wie könnte eine mögliche Lösung des Problems aussehen? (nicht programmieren!)

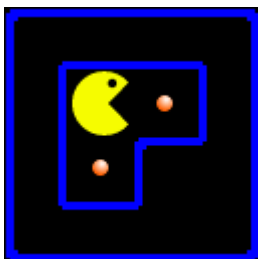


Abbildung 1: Die gewünschte Welt für Aufgabe 3.