

Organic Computing

PAC-MAN



Langzeitaufgabe
OC WS 2011/2012

Organic Computing

- Pac-Man -

Programmieraufgabe

- Aufgabestellung
 - Entwicklung eines Pac-Man und eines Geister-Teams
 - Umsetzung durch Methoden des Organic Computing
 - Agenten sollen automatisch, eigenständig und intelligent sein

Organic Computing

- Pac-Man -

Programmieraufgabe

- Ziel
 - Kennenlernen und selbständiges Erforschen von Techniken, die in Organic Computing verwendet werden

Organic Computing

- Pac-Man -

Programmieraufgabe

- Organisatorisches
 - Umsetzung in 3er Teams, wenn möglich
 - Abgabe spätestens am 17.01.12
 - Abgabe per Mail (Geister-, Pac-Man-Klasse)
 - Challenge: 24.01.2011
 - Wettbewerb und Siegerehrung
 - Das Spiel: Eclipse-Projekt (Datei Pac-Man.zip)

Organic Computing

- Pac-Man -

Programmieraufgabe

- Installation
 - Euer Pac-Man muss in das package **game.player.pacman** und die abstrakte Klasse **gui.AbstractPlayer** implementiert werden
 - Euer Geister-Team muss in das package **game.player.ghost** und die abstrakte Klasse **gui.AbstractGhost** implementiert werden

Organic Computing

- Pac-Man -

Programmieraufgabe

- Installation
 - Eure Pac-Man-Klasse muss die Methode `public int getAction(Game game, long timeDue){}` enthalten
 - Eure Geisterklasse muss die Methode `public int[] getActions(Game game, long timeDue){}` enthalten
 - `GetAction()`-Methoden verlangen als Rückgabewert die Richtung in die euer Pacman/Geist gehen soll:
Up - Right - Down - Left -> 0 - 1 - 2 - 3

Organic Computing

- Pac-Man -

Programmieraufgabe

- Was sollen Pac-Man/Geister können?
 - Pac-Man soll Geistern ausweichen, schnell und effizient Pillen fressen und möglichst viele Geister besiegen
 - Geister sollen Pac-Man im Weg behindern, ihn jagen und seine Leben reduzieren
 - Informationen über die Spielwelt können über die Variable game erfahren werden
 - Durch auswerten dieser Informationen sollen Pac-Man/Geister entscheiden, welche Richtung sie einschlagen

Organic Computing

- Pac-Man -

Programmieraufgabe

- Zielsetzung
 - Organic Computing Techniken und Werkzeuge erlernen und anwenden
 - Es wird erwartet, dass ihr...
 - ... herausfindet, welche Techniken geeignet sind
 - ... euch begründet für eine Technik entscheidet
 - ... diese umsetzt und damit einen „intelligenten Spieler“ entwickelt

Organic Computing

- Pac-Man -

Programmieraufgabe

- Wettbewerb
 - Am Ende des Semesters soll jede Gruppe ihre Pac-Man/Geister und die verwendeten Techniken kurz vorstellen
 - Welches Verfahren habt ihr euch angesehen
 - Wie funktionieren eure Spieler
 - Wie war die Aufgabenverteilung in der Gruppe
 - Danach treten die Gruppen gegeneinander an
 - Die Gewinnergruppe erhält einen Preis

Organic Computing

- Pac-Man -

Programmieraufgabe

- Hinweis
 - Cheating ist nicht erlaubt
 - Offizielle Seite: <http://www.pacman-vs-ghosts.net/>
- Ansprechpartner bei Fragen aller Arten
 - Hendrik Reuter (hen_reuter@web.de)
 - Mustafa Zengin (mustafazengin@msn.com)

Organic Computing

- Pac-Man -

Programmieraufgabe

- Über das Interface Game können Information zum Spielzustand gesammelt werden
- Das Spielfeld ist in Zellen unterteilt. Jede Zelle ist mit einer eindeutigen ID vermerkt welche von linken oberen ecke beginnt und nach rechts unten hin hochzählt.
- Spieler, Geister, Pills sowie Powerpills befinden sich auf solchen Zellen.
- Positionen werden durch die Ids eindeutig

Organic Computing

- Pac-Man -

Vorgegebene Funktionen in Games

- Methoden zum Abfragen der Positionen für Geister, Pac-Man, Pills und PowerPills
- Abfragemöglichkeit der Zustände der Geister (Inaktive Zeit etc.) und Pac-Man
- Methoden zum Abfragen der Routen von Start- und Zielknoten, nächst mögliche Knoten
- Distanzalgorithmen (Euklid, Djekstra) zum berechnen von kürzeren Pfaden

Organic Computing

- Pac-Man -

Funktion für Position

- checkPill, checkPowerPill – Abfrage ob noch Pillen/Powerpillen auf der Position liegen
- getPillIndicesActive, getPowerpillIndicesActive – Info über die Position von noch nicht gesammelten Pillen
- getCurGhostLoc, getCurPacManLoc – Info über aktuelle Position von Geistern und Pac-Man

Organic Computing

- Pac-Man -

Funktion für Routen

- getPacManNeighbours – gibt Mögliche Laufrichtungen an
- getRevers(direction) – gibt die Gegenrichtung an
- getNextPacManDir – gibt nächst Richtung zum Ziel an
- getPathDistanz, getManhattanDistanz, getEuclidDistanz – gibt Distanzwerte zwischen zwei Knoten an
- Alle Funktion stehen für die Geister und für den Pac-Man zur Verfügung

Organic Computing

- Pac-Man -

Funktion für Zustände der Spieler

- getCurGhostDir, getCurPacManDir – Die aktuelle Richtung in welche die Spieler laufen
- isEdible, getEdibleTime – Information zu Inaktivem Zustand der Geister
- getLairTime – Information zum Aufenthaltszeit im Gefängnis der Geister
- ... Es stehen 55 Funktion zu Verfügung um sich ein Bild über die Spielsituation zu machen

Organic Computing

- Pac-Man -

Beispiel für ein Pac-Man

- Idee: Pac-Man soll mit einer Wahrscheinlichkeit von den Geistern wecklaufen oder zu der nächst näheren Pille gehen. Alle 2 Sekunden soll Pac-Man sich vom neuen Entscheiden

Organic Computing

- Pacman -

Training

- Training ist über die GUI Auswahl möglich
- Nach den Anzahl an Training wird das letzte Spiel in der GUI dargestellt
- Indikatoren zum Messen der Fitness können Punktestände, Zeiten für ein Level oder für das gesamte Spiel sein.