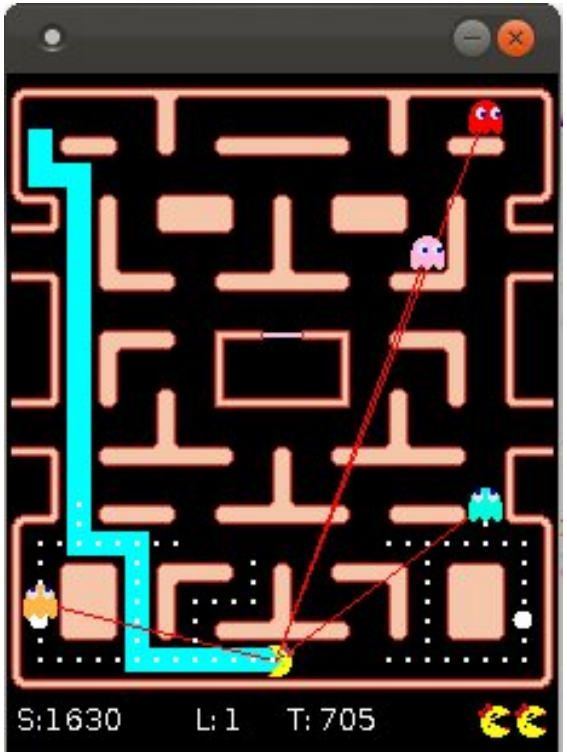
Pac-Man-Challenge



Langzeitaufgabe

Übung Organic Computing - Pac-Man-Challenge

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

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

Übung Organic Computing - Pac-Man-Challenge

- Organisatorisches
  - Umsetzung in 3er Teams
  - Wir überprüfen auf Plagiate der Vorjahre
  - Abgabe **spätestens** am **26.01.2018**
  - Abgabe per Mail (Geister-, Pac-Man-Klasse) an shuka@sra.uni-hannover.de
  - Challenge: 02.02.2018
  - Wettbewerb und Siegerehrung
  - Das Spiel: Eclipse-Projekt (Datei Pac-Man.zip in Stud.IP)

Installation

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

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 autonom entscheiden, welche Richtung sie einschlagen.

Zielsetzung Pac-Man-Challenge

- 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

Übung Organic Computing - Pac-Man-Challenge

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

Übung Organic Computing- Pac-Man-Challenge

- Hinweis
  - Cheating ist nicht erlaubt
  - Täuschungsversuche führen zu NB in der gesamten VA

Übung Organic Computing - Pac-Man-Challenge

- 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.

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

Funktionen 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

Funktion für Routen

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

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

Beispiel für ein Pac-Man

- Neuronale Netze
- LCS
- Regressionslearning

Training

- Training ist über die GUI Auswahl möglich.
- Nach der 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.

Code-Konventionen

- Eure Pac-Man-Klasse: PacmanGroupX
- Eure Ghost-Klasse: GhostGroupX
- Trainingsdaten: GroupXDataForLevelY

- X : Eure Gruppennummer z.B. PacmanGroup1 bzw. GhostGroup1
- Y: Game Level z.B. Group1DataForLevel3

DEMO

Quelle: http://www.fightersgeneration.com/nx/chars/pacman/pacman-thumbsup.jpg