

1. Reaktive Agenten und Multiagentensysteme

„Reaktive (bzw. subkognitive) Agenten verfügen nicht über eigenes Wissen, sondern agieren nur aufgrund ihrer Wahrnehmung direkt und ohne Entscheidungsprozess. Der reaktive Agent erhält Sensorinformationen und wählt aufgrund von Bedingungs-Aktions-Regeln eine Aktion aus.“

(Quellen „Wikipedia“ <http://de.wikipedia.org/wiki/Software-Agent>)

Wichtige Eigenschaften:

autonom – das Programm arbeitet weitgehend unabhängig von Benutzereingriffen

reaktiv – das Programm reagiert auf Änderung der Umgebung

sozial – das Programm kommuniziert mit anderen Agenten

„Bei einem Multiagentensystem oder MAS handelt es sich um ein System aus mehreren gleichartigen oder unterschiedlich spezialisierten handelnden Einheiten, die kollektiv ein Problem lösen.“ (Quellen „Wikipedia“ <http://de.wikipedia.org/wiki/Multiagentensystem>)

2. Lernziel

Das Lernziel besteht darin den Teilnehmern einen Einblick in das Design und die Implementierung primitiver Verhaltensweisen zu geben welche im Zusammenspiel ein komplexes Verhalten erzeugen. Das Projekt endet mit einer kurzen Präsentation der erstellten Regeln und einer Reflexion der entwickelten Verhaltensweisen im Kontext eines Turniers.

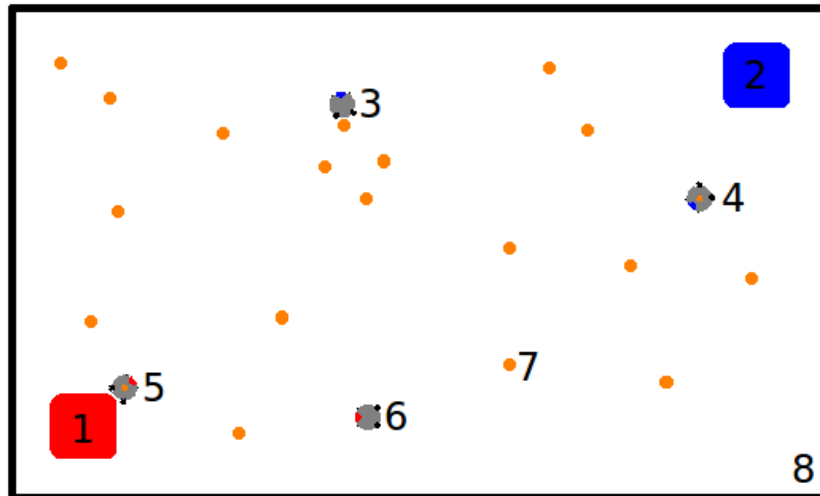
3. Ablauf

Es werden Gruppen von ca. 2-3 Personen gebildet. Die Gruppen entwickeln und programmieren anschließend für ein zuvor definiertes Spiel geeignete Regeln für Agenten die am Ende der Entwicklungsphase in einer Art Turnier gegen einander antreten.

4. Spiel

Zwei unterschiedliche Roboterteams erobern einen für sie unbekannten Planeten. Ihre Mutterschiffe sind bereits gelandet und sie erkundschaften die Umgebung und sammeln in Konkurrenz über den Planeten zufällig verstreute Ressourcen ein die sie anschließend zum Mutterschiff zurück bringen. Ziel des Spiels ist es mit dem im Mutterschiff vorhandenen Treibstoff so viele Ressourcen wie möglich einzusammeln und zum Mutterschiff zu befördern. Die Roboter selber verbrauchen den Treibstoff und müssen diesen Regelmäßig beim Mutterschiff wieder auftanken. Andererseits bleiben sie auf dem Spielfeld ohne Funktionalität zurück. Zum einsammeln der Ressourcen wird eine gewisse Zeit benötigt, zudem sind Roboter die eine Ressource zum Mutterschiff transportieren langsamer als normal. Das Spiel endet wenn der Treibstoff beider Teams verbraucht ist.

5. Visualisierung in 2D



1. Mutterschiff Team Rot
2. Mutterschiff Team Blau
3. Roboter sammelt Resource
4. Roboter transportiert Resource
5. Roboter liefert Resource ab
6. Roboter sucht Ressourcen
7. Eine Resource
8. Das Spielfeld (Der Planet)

6. Optionale Zusatzfunktionalitäten

- Roboter die auf dem Spielfeld bedingt durch Treibstoffmangel zurück bleiben können durch andere Teamroboter eine Treibstoffspende erhalten und somit zum auftanken zum Mutterschiff zurück kehren.
- Roboter können das Feindliche Mutterschiff angreifen so das dieses anfängt zu brennen und der Treibstoffgehalt stetig sinkt. Die Teamroboter werden über ihr brennendes Mutterschiff informiert und können zu zweit das Mutterschiff wieder löschen.
- Roboter können feindliche Roboter attackieren sofern sich diese in ihrer unmittelbaren Nähe befinden. Diese sind dann für eine kurze Dauer außer Funktion.
- Roboter können Ressourcen von feindlichen Robotern klauen sofern diese zuvor außer Funktion gesetzt wurden.

7. Informationen die den Robotern zur Verfügung stehen

- Mögliche Kollision mit einer Wand.
- Position des eigenen Mutterschiffs.
- Position des feindlichen Mutterschiffs in unmittelbarer Nähe.
- Position der Teamroboter.
- Positionen von feindlichen Robotern in unmittelbarer Nähe.
- Position von Ressourcen in unmittelbarer Nähe oder in der Nähe von Teamrobotern.
- Teamroboter ohne Funktionalität bezüglich eines Angriffes oder Treibstoffmangels.
- Eigener Treibstoffgehalt
- Angriff des Mutterschiffs.
- Brennen des Mutterschiffs.

8. Aktionen die die Robotern ausführen können

- Drehe um Winkel X.
- Gehe vorwärts.
- Gehe in Richtung der nächstliegenden bekannten Ressource.
- Gehe in Richtung des eigenen Mutterschiffs.
- Gehe in Richtung des gesichteten feindlichen Mutterschiffs.
- Gehe in Richtung eines ohne Treibstoff liegen gebliebenen Teamroboters.
- Sammele Ressource ein.
- Attackiere feindlichen Roboter.
- Attackiere feindliches Mutterschiff.
- Lösche eigenes Mutterschiff.
- Treibstoff tanken am Mutterschiff.
- Einen nächstliegenden Teamroboter auftanken.

9. Regelimplementation

Die Gruppen erhalten ein Spielframework welches die grafische Oberfläche sowie alle essentiellen Funktionalitäten für das Spiel bereit stellt. Informationen wie oben gelistet können abgefragt und entsprechende Aktionen ausgeführt werden. Zudem können die Gruppen ihre Agenten gegen ein primitives vorhandenes Team oder gegen sich selber antreten lassen, um die bereits implementierte Funktionalität zu überprüfen, diese zu erweitern oder gegebenenfalls zu optimieren.

10. Regelbeispiel:

Priorität	Regel	Aktion
4	Wand vor Agent	drehe um 180°+zufällige Varianz
3	Trage Ressource & @ Mutterschiff	gebe Ressource Mutterschiff
2	Trage Ressource	gehe Richtung Mutterschiff
1	Ressource Gesichtet	sammele Ressource ein
0	Ist immer wahr	gehe geradeaus

Der Algorithmus wählt zunächst die Regel mit der höchsten Priorität aus. Ist die Regel wahr so wird die zugehörige Aktion ausgeführt. Ist die Regel nicht wahr so wird der Prozess iterativ mit der Regel, welche die nächst geringeren Priorität besitzt, fortgeführt bis eine wahre Regel gefunden wird.

Weiterführende Informationen zum Thema

<http://de.wikipedia.org/wiki/Software-Agent>

<http://de.wikipedia.org/wiki/Multiagentensystem>