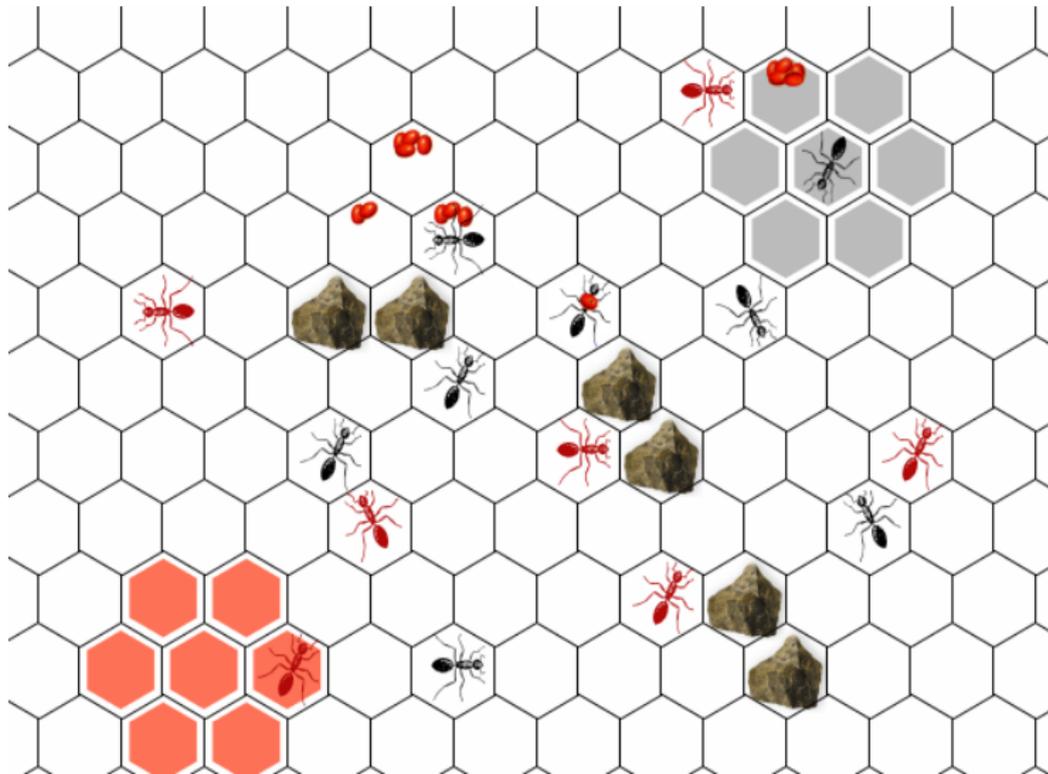


# Das Projekt

# Die Ameisenart *formica artificiosa*

Ameisen dieser Art in ihrer natürlichen Umgebung



# Hexagonale Zellen

## Jede Zelle

- kann ein Hindernis enthalten (vielleicht einen Stein oder Wasser, aber das ist nicht näher spezifiziert).
- kann *eine* Ameise enthalten, wenn sie nicht schon ein Hindernis enthält.
- kann eine beliebige Anzahl von Futterstücken enthalten.
- kann ein Teil des Ameisenhügels eines bestimmten Ameisenstammes sein.
- kann Duftspuren enthalten. Jeder Ameisenstamm hat sechs verschiedene Duftmarker, von dem jeder entweder in der Zelle vorhanden ist oder nicht.

# Ameisen

## Ameisen

- verhalten sich völlig autonom.
- kommunizieren und orientieren sich mit Duftmarkern.
- sind in Ameisenstämmen organisiert.

## Ameisenstämme

- stehen im Wettbewerb um Futter.
- werden mit (hoffentlich) intelligentem Design entworfen.

# Praktikumsinhalt

Wir wollen die Ameisenart *formica artificiosa* in Java simulieren und Programme schreiben

- zur Entwicklung und Erprobung von Ameisenstämmen.
- zum Ausrichten von Wettbewerben zwischen Ameisendesignern.

# Ein Wettbewerb für Ameisendesigner

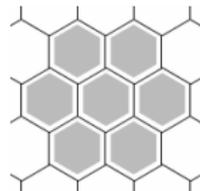
Das Ziel ist, einen Ameisenstamm zu entwickeln, der in allen Situationen möglichst viele Futterstücke findet und zurück zu seinem Ameisenhügel bringt.

- Spieler entwickeln Ameisenstämme.
- Ameisenstämme treten gegeneinander auf einem Spielfeld an.
- Gesamtgewinner wird in einem Turnier ermittelt.

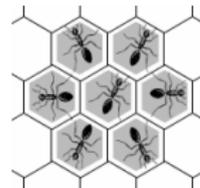
# Spielablauf I

Die Spieler entwickeln ihre Ameisenstämme und reichen sie dann auf einem Spielserver ein.

- Bekannt ist nur, wie groß der Ameisenhaufen ist, aber nicht, wie das Spielfeld aussieht.



- Spieler legt fest, wie seine Ameisen auf den Haufen zu setzen sind.



- Spieler bestimmt das Verhalten der Ameisen, indem er das Gehirn jeder Ameise „programmiert“.

## Spielablauf II

Wenn alle Spieler ihre Ameisenstämme eingereicht haben, dann kann man verschieden Stämme gegeneinander antreten lassen.

- ① Setze Ameisenstämme auf verschiedene Ameisenhaufen desselben Spielbretts.
  - Üblicherweise spielen jeweils zwei Stämme gegeneinander, aber auf größeren Spielfeldern sind auch mehr möglich.
  - Alle Ameisenhaufen sind gleich groß.
    - ⇒ Rückspiele mit vertauschten Ameisenhügeln möglich
- ② Lasse die Ameisen eine bestimmte Zeit lang fleißig arbeiten.
- ③ Gewonnen hat, wer danach die meisten Futterstücke auf seinem Hügel liegen hat.

# Praktikumsinhalt

Wir wollen die Ameisenart *formica artificiosa* in Java simulieren und Programme schreiben

- zur Entwicklung und Erprobung von Ameisenstämmen.
- zum Ausrichten von Wettbewerben zwischen Ameisendesignern.

# Entwicklung von Ameisenstämmen

Gehirne werden in der Programmiersprache ANTBRAIN programmiert.

```
lookingForFood = 1;
while true do {
  if sense(here, food) then {pickup; lookingForFood = 0} else skip;
  if sense(here, home) then {drop; lookingForFood = 1} else skip;
  t = rand%2 + rand%2 - 1; turn(t);
  walk
}
```

Die Ameisen eines Ameisenstammes sind gegeben durch:

- ihre Position auf dem Ameisenhaufen
- ein ANTBRAIN-Programm, welches ihr Verhalten bestimmt

# Entwicklung von Ameisenstämmen

## Programmiersprache ANTBRAIN

- Minimalanforderung
- Serverkompatibilität
- einfach zu implementieren, dafür nicht sehr bequem

## Ameisenentwicklung

- Werkzeuge zum Experimentieren mit Ameisen
- Erweiterte Sprachen, die in ANTBRAIN zurückübersetzt werden
- ...

# Simulation der Ameisen im Spielservers

Wenn Ameisenstämme gegeneinander antreten passiert Folgendes:

- 1 Die Ameisenstämme werden auf Ameisenhaufen gesetzt und die Gehirne initialisiert.
- 2 Die Ameisen bewegen sich dann selbständig, indem sie die Programme in ihren Gehirnen abarbeiten.
  - Zeit ist durch die Anzahl der Spielrunden beschränkt (Größenordnung: 1 Million).
  - In jede Spielrunde kommen die Ameisen nacheinander dran und machen je genau einen Schritt in ihrem ANTBRAIN-Programm.
  - Durch Instruktionen wie `walk` oder `turn(i)` können sich die Ameisen auf diese Art bewegen.
- 3 Am Ende wird ermittelt, wie viele Futterstücke jeder Ameisenstamm auf seinem Ameisenhaufen liegen hat.

# Detailinformationen

Siehe Spezifikation.

# Aufgabe

Entwicklung zweier getrennter Komponenten, die über eine Netzwerkverbindung miteinander kommunizieren können.

## ① Der Server

- speichert Spielfelder.
- erlaubt den Spielern Ameisenstämme einzureichen.
- berechnet, was passiert wenn verschiedene Ameisenstämme auf einem Spielfeld gegeneinander antreten.
- speichert Spiele zur späteren Ansicht.

## ② Der Client

- erlaubt das Editieren und Ausprobieren von Ameisenstämmen.
- kann Ameisenstämme bei einem Server einreichen.
- zeigt die auf einem Server liegenden Spiele grafisch an.

Mehrere Spieler können gegeneinander antreten, indem sie sich mit ihre Clients mit demselben Server verbinden.