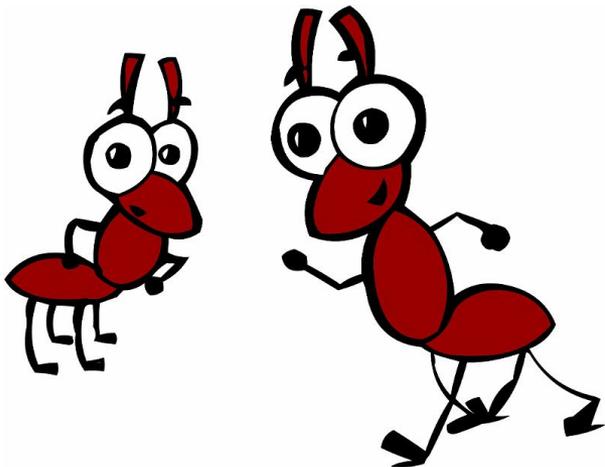


Ameisensimulator 2.0

Weiterentwicklung eines Demonstrators
für die Navigationsgeräte der Zukunft



Agenda

- > Schwarmintelligenz (25')
- > Ausgangspunkt (5')
- > Softwareentwicklung (30')

Agenda

- > Schwarmintelligenz (25')
 - Was ist das?
 - Kleines Beispiel
 - Berechnung & Unberechenbarkeit
 - Ziele und Ausblicke

- > Ausgangspunkt (5')

- > Softwareentwicklung (30')

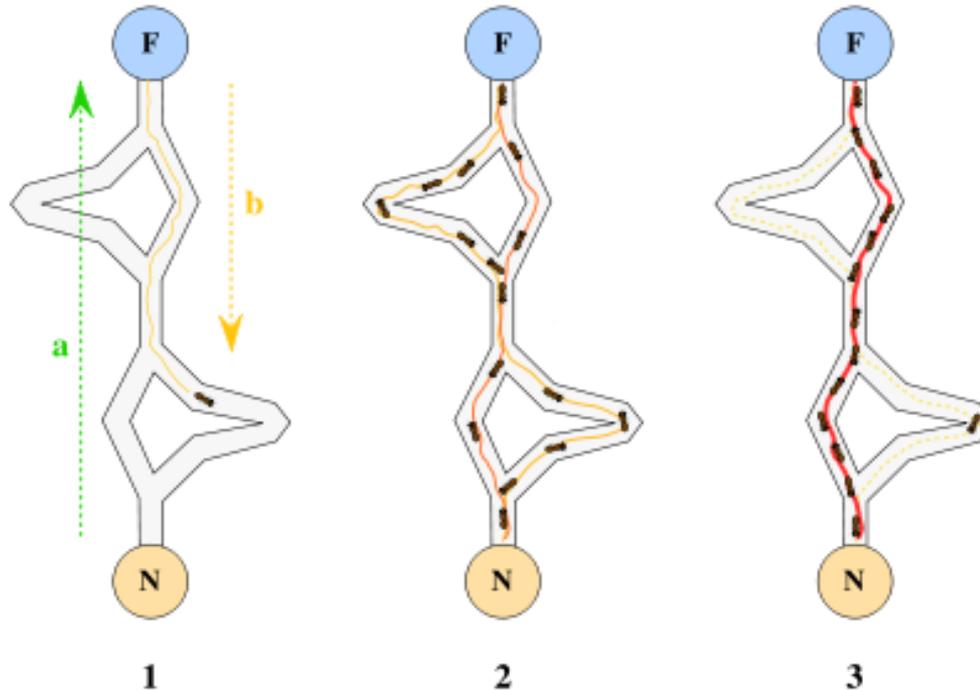
Agenda

- > Schwarmintelligenz (25')
 - Was ist das?
 - Kleines Beispiel
 - Berechnung & Unberechenbarkeit
 - Ziele und Ausblicke

- > Ausgangspunkt (5')

- > Softwareentwicklung (30')

Schwarmintelligenz Was ist das?



- 1.) Die erste Ameise bahnt sich ihren Weg vom Nest zum Futter und zurück. Auf Ihrem Rückweg hinterlässt sie Pheromone.
- 2.) Die nächsten Ameisen folgen ihr auf 4 verschiedenen Wegen. Die Pheromonspur nimmt pro Ameise weiter zu.
- 3.) Fast alle Ameisen folgen dem kürzesten Weg. Wenige Ausreißer testen weiterhin auch andere Wege.

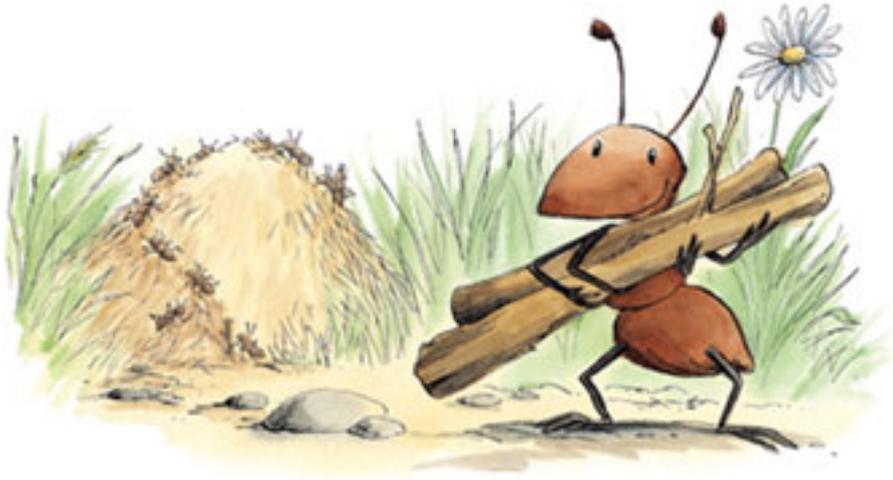
Agenda

- > Schwarmintelligenz (25')
 - Was ist das?
 - **Kleines Beispiel**
 - Berechnung & Unberechenbarkeit
 - Ziele und Ausblicke

- > Ausgangspunkt (5')

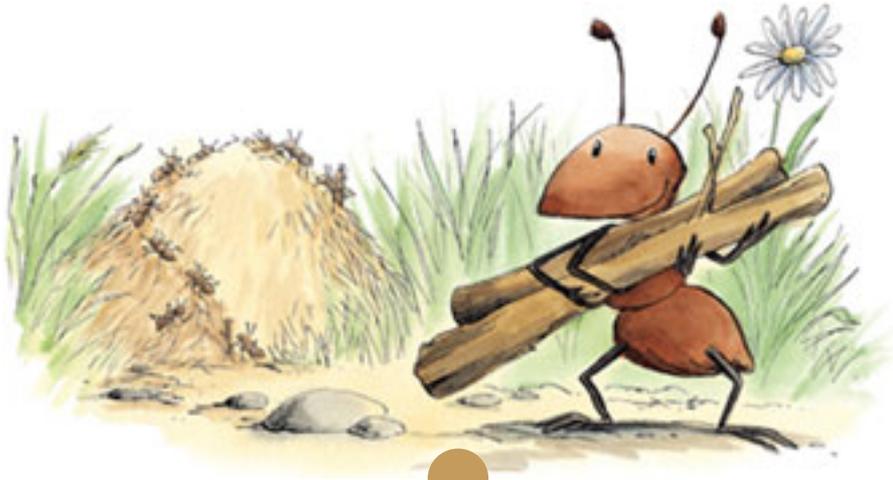
- > Softwareentwicklung (30')

Schwarmintelligenz



Fred soll für die Ameisenkönigen
die Futtersuche organisieren

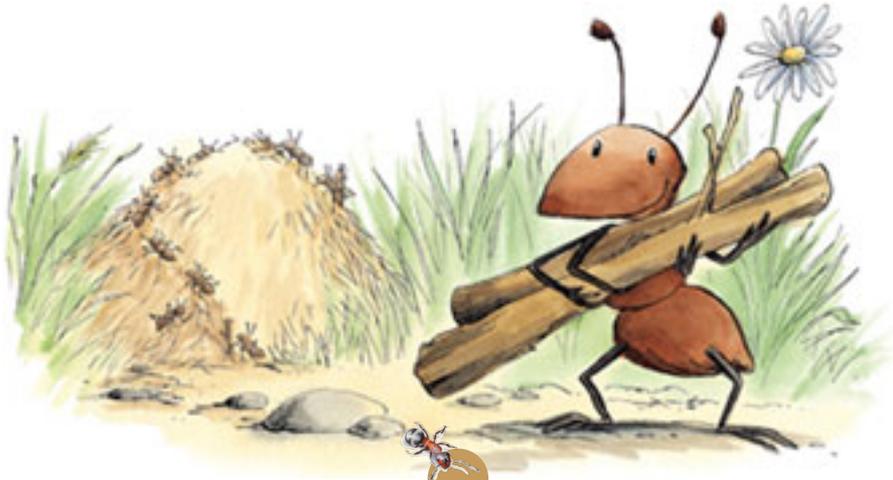
Schwarmintelligenz



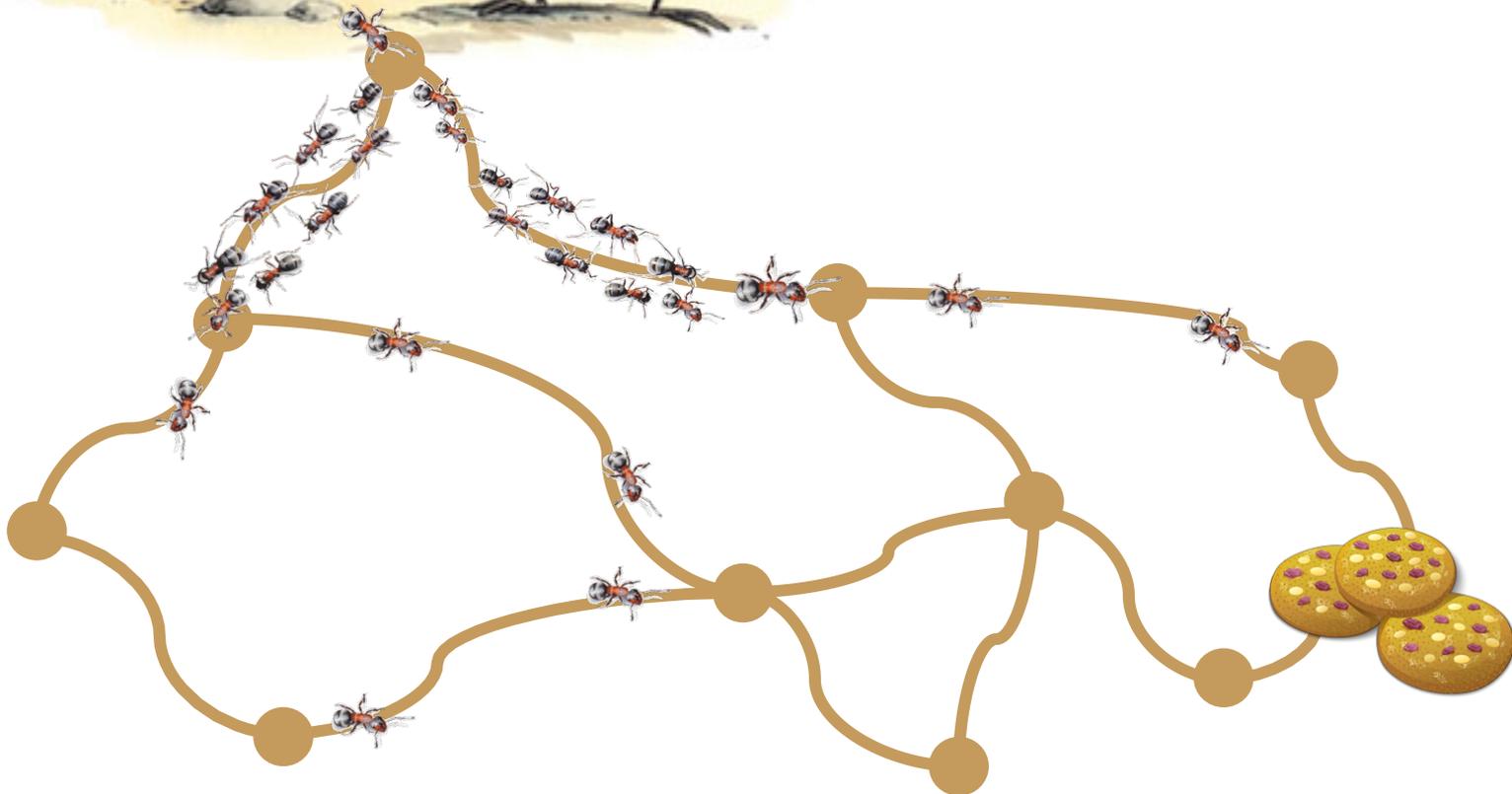
Fred kennt die Umgebung und weiß,
dass irgendwo am anderen Ende Kekse liegen



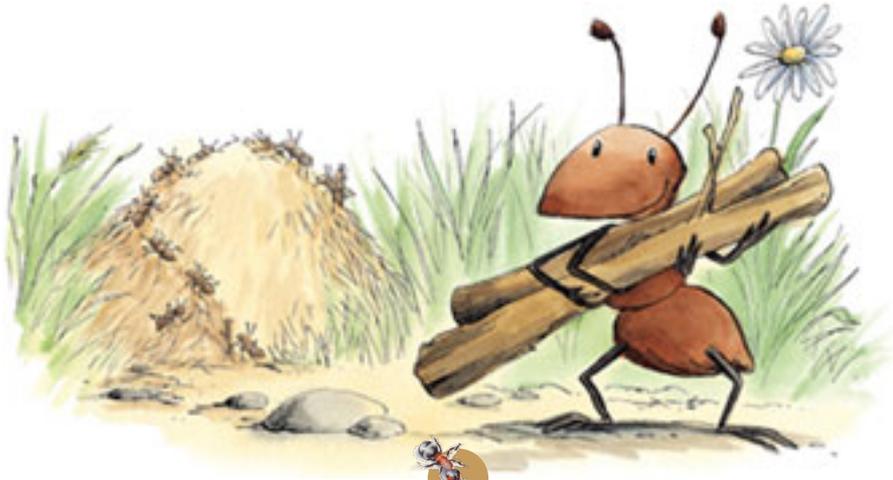
Schwarmintelligenz



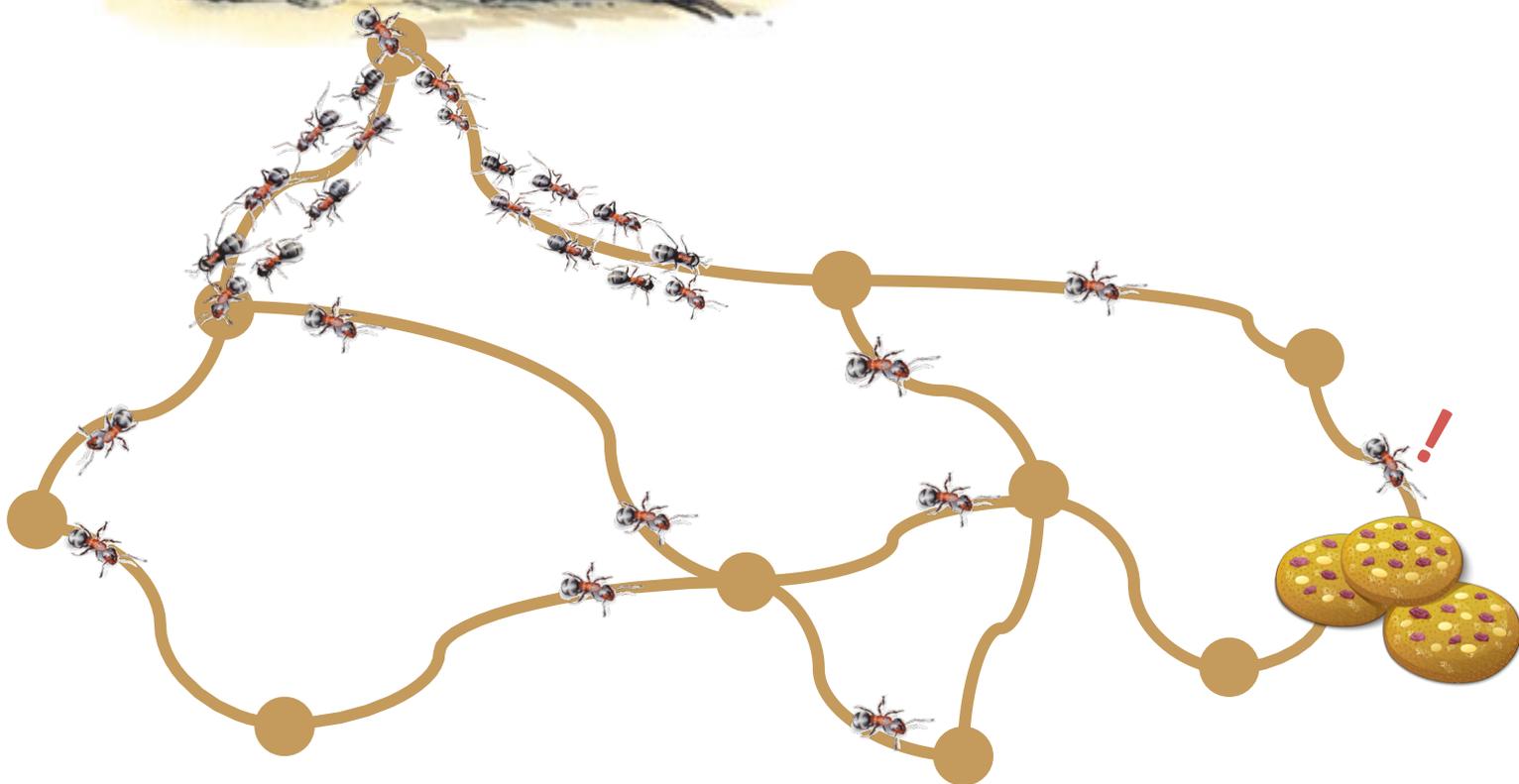
So schickt Fred seine Ameisenfreunde los, um sich auf die Suche zu machen ...



Schwarmintelligenz



Sie verstreuen sich, bis die erste Ameise die Kekse findet

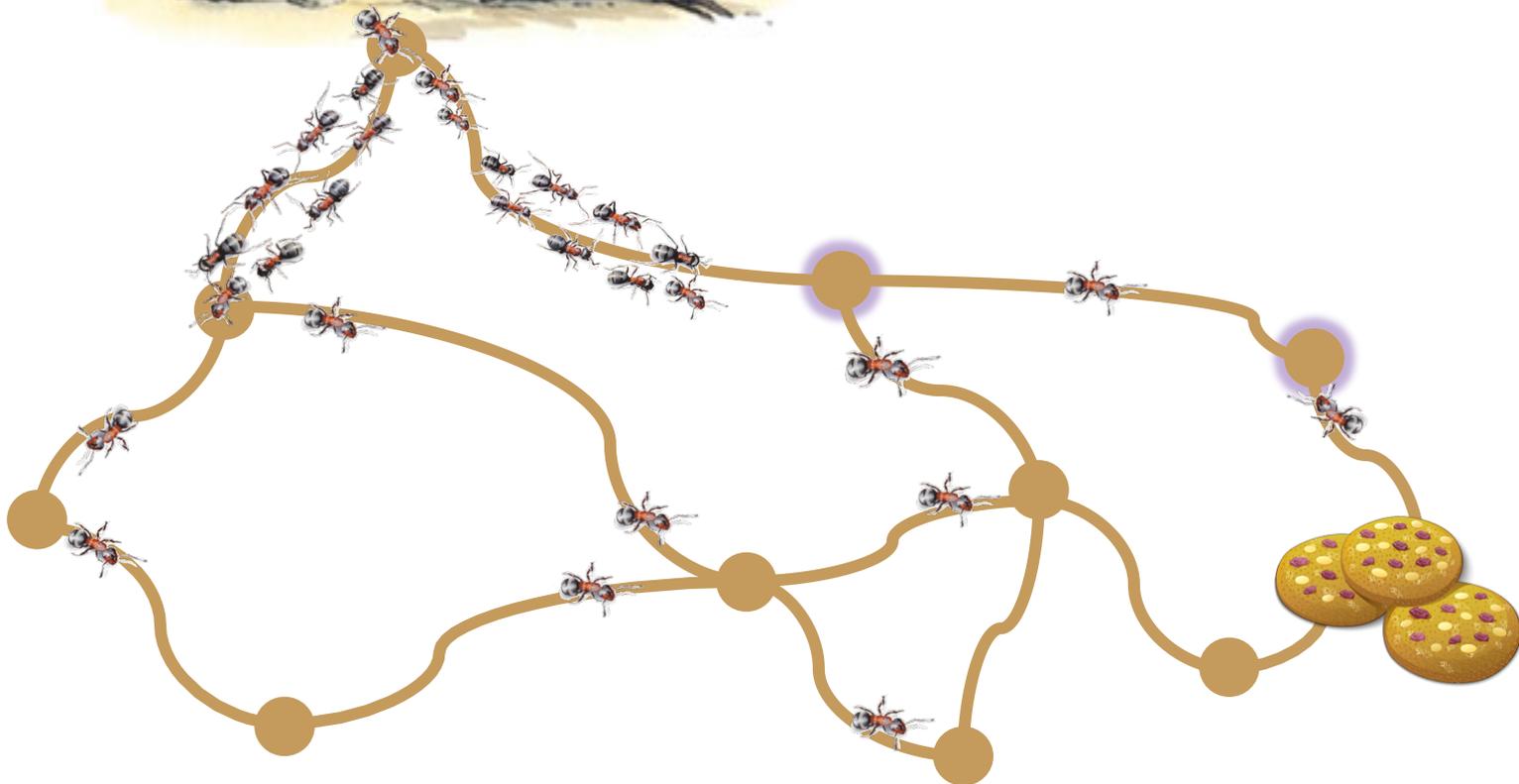


Schwarmintelligenz



Nachdem die Ameise das Futter gefunden hat, macht sie sich auf gleichem Weg nach Hause.

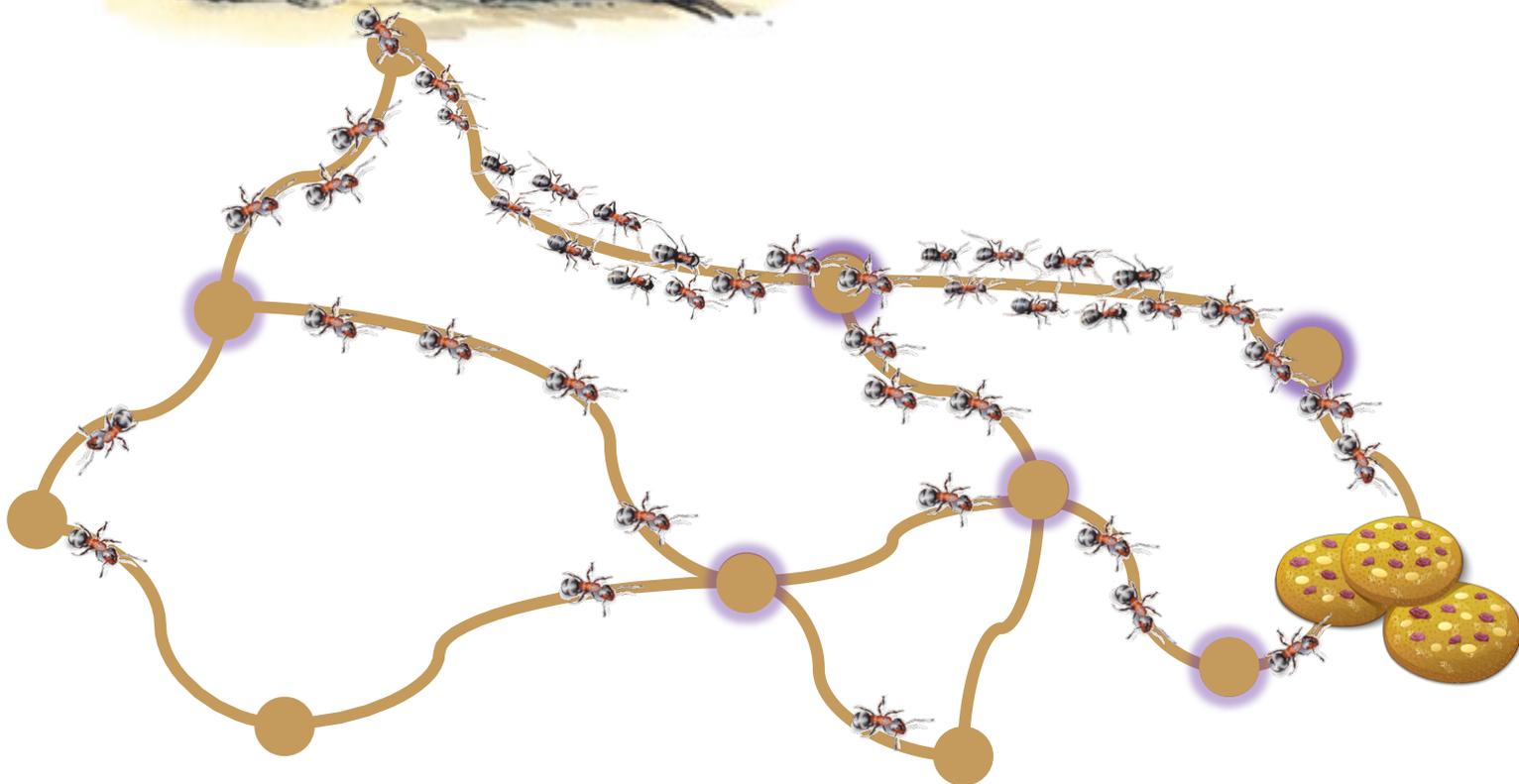
Um ihren Kollegen zu zeigen, dass es auf diesem Weg Futter gibt, legt sie Pheromone aus.



Schwarmintelligenz



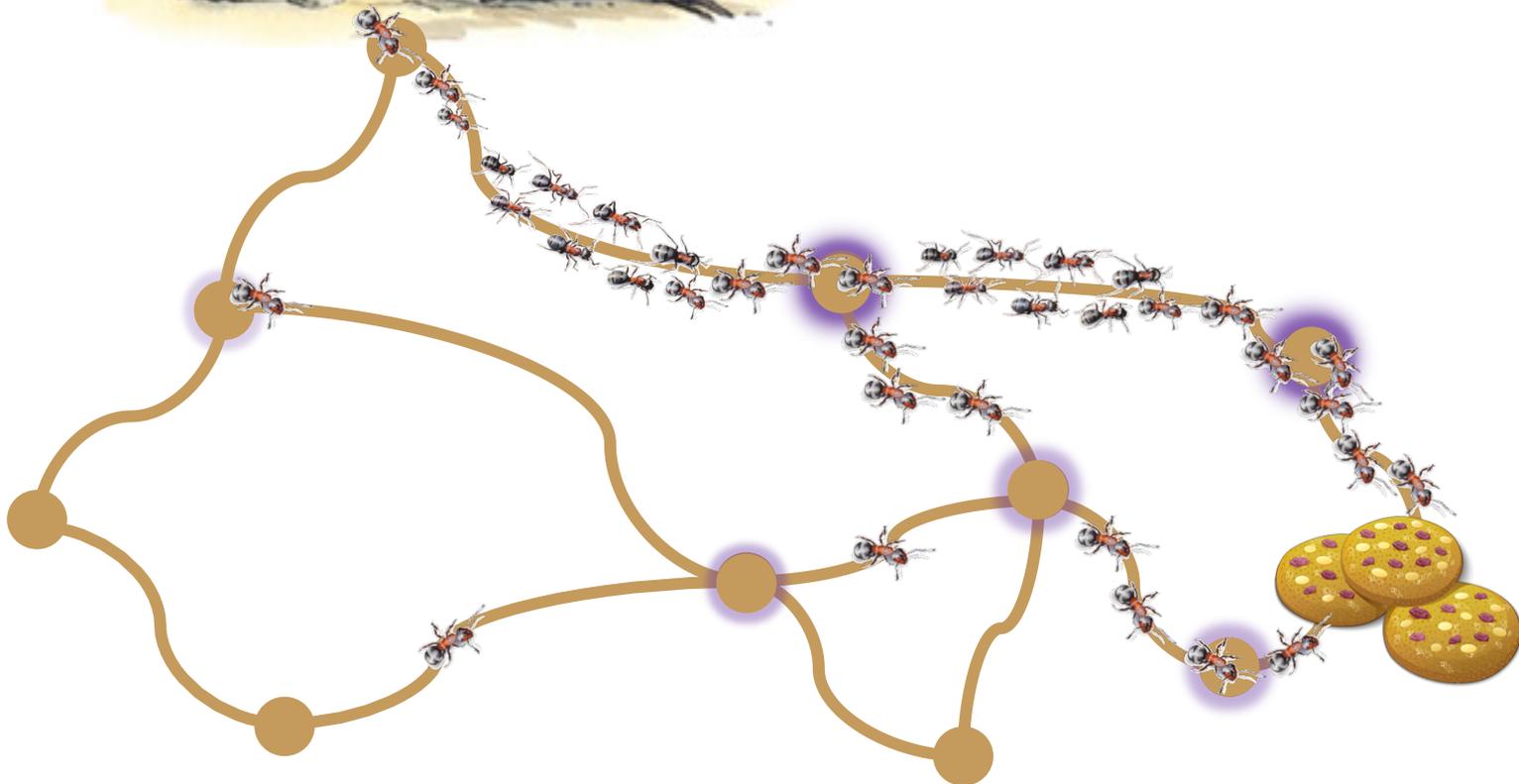
Je mehr Ameisen der Pheromonspur folgen, desto intensiver wird diese.



Schwarmintelligenz



Schließlich kristallisiert sich ein Pfad heraus, dem dann fast alle Ameisen folgen.



Agenda

- > Schwarmintelligenz (25')
 - Was ist das?
 - Kleines Beispiel
 - **Berechnung & Unberechenbarkeit**
 - Ziele und Ausblicke
- > Ausgangspunkt (5')
- > Softwareentwicklung (30')

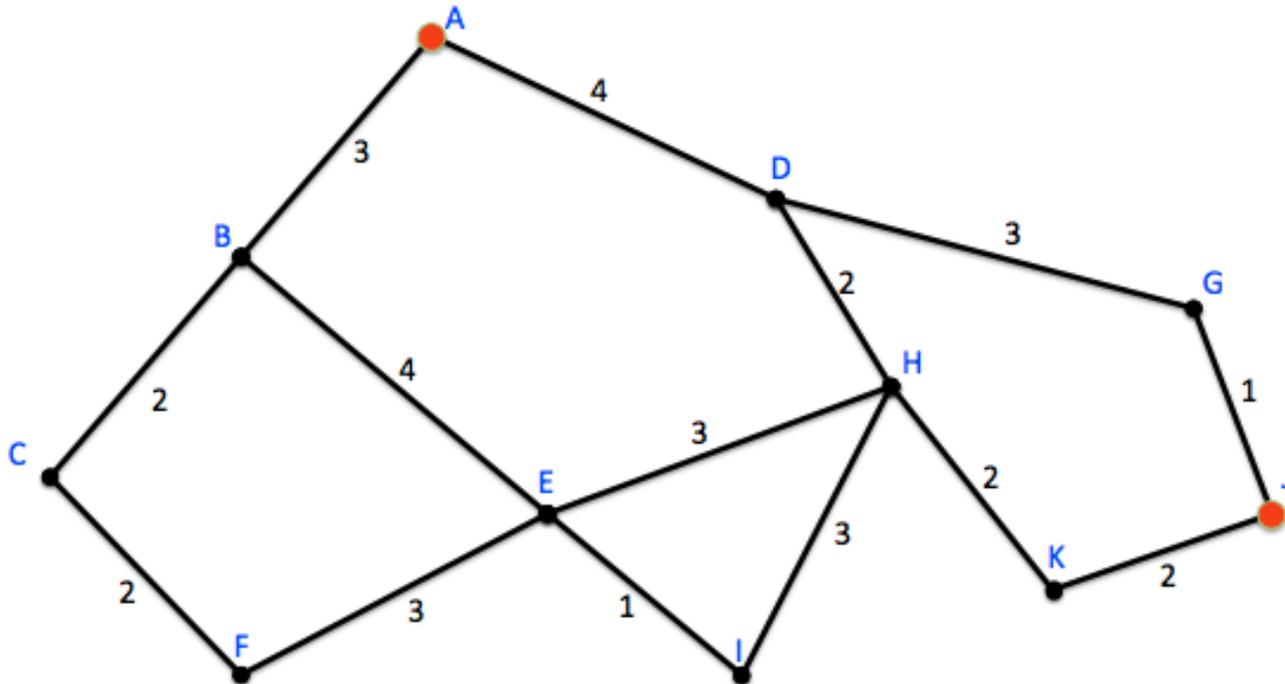
Schwarmintelligenz

Der Ablauf als Pseudocode

- > Aufstellen des Graphen
- > Aufstellen einer Wahrscheinlichkeitstabelle für jeden Knoten
- > Aufstellen der mathematischen Formeln
- > Wiederhole:
 - Zufallswahl eines Ameisenweges
 - Anpassen der Wahrscheinlichkeiten der Tabelle jedes passierten Knotens
- > Bis Wegstrecke eindeutig

Schwarmintelligenz Der Ablauf als Pseudocode

- > Aufstellen des Graphen
- > Aufstellen einer Wahrscheinlichkeitstabelle für jeden Knoten
- > Aufstellen der mathematischen Formeln
- > Wiederhole:
 - Zufallswahl eines Ameisenweges
 - Anpassen der Wahrscheinlichkeiten der Tabelle jedes passierten Knotens
- > Bis Wegs



Schwarmintelligenz Der Ablauf als Pseudocode

- > Aufstellen des Graphen
- > Aufstellen einer Wahrscheinlichkeitstabelle für jeden Knoten
- > Aufstellen der mathematischen Formeln
- > Wiederhole:
 - Zufallswahl eines Ameisenweges
 - Anpassen der Wahrscheinlichkeiten der Tabelle jedes passierten Knotens
- > Bis Wegstrecke eindeutig

A	B	D
J	0,5	0,5

B	C	E	A
J	0,333	0,333	0,333

C	F	B
J	0,5	0,5

... für jeden Knoten

Schwarmintelligenz Der Ablauf als Pseudocode

- > Aufstellen des Graphen
- > Aufstellen einer Wahrscheinlichkeitstabelle für jeden Knoten
- > Aufstellen der mathematischen Formeln
- > Wiederhole:
 - Zufallswahl eines Ameisenweges
 - Anpassen der Wahrscheinlichkeiten der Tabelle jedes passierten Knotens
- > Bis Wegstrecke eindeutig

Verdampfungskoeffizient

$$\Delta P_{s,d} = \frac{c_1}{t_{s,d}} + c_2$$

Verdampfung der Pheromone

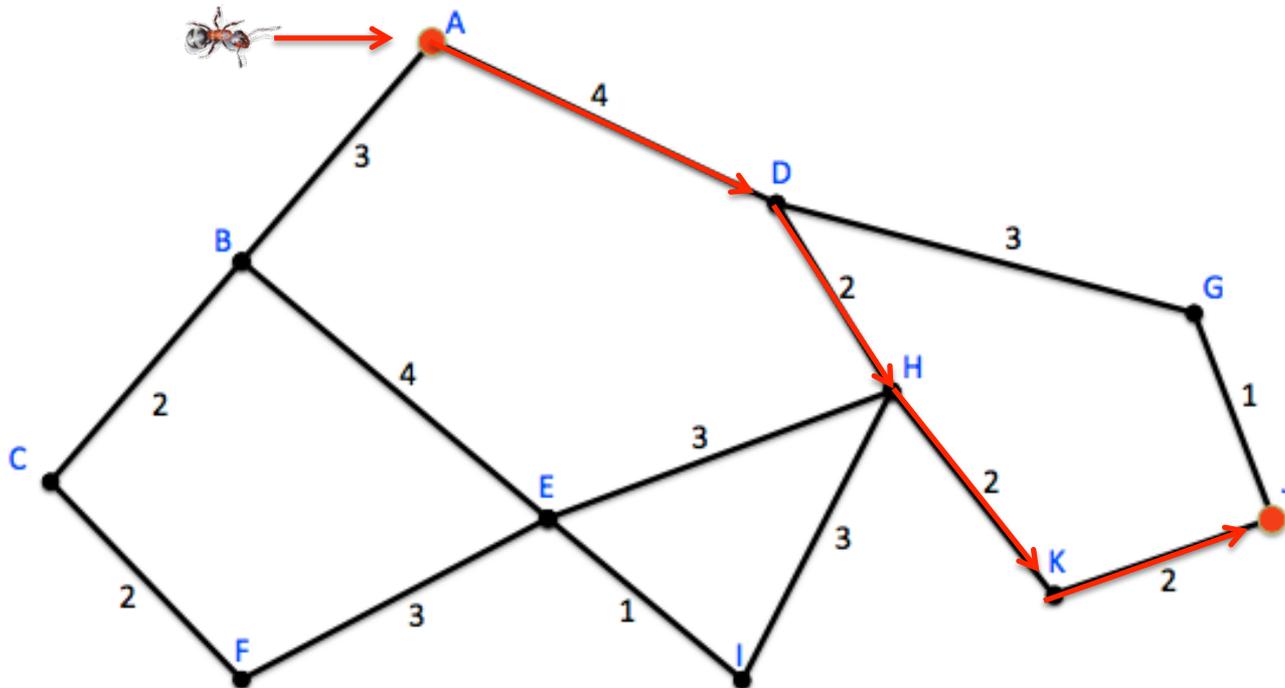
$$P_{d,i} = \frac{P_{d,i}}{1 + \Delta P_{s,d}} \quad \forall i \neq f$$

Erhöhung der Pheromon-Konzentration

$$P_{d,f} = \frac{P_{d,f} + \Delta P_{s,d}}{1 + \Delta P_{s,d}}$$

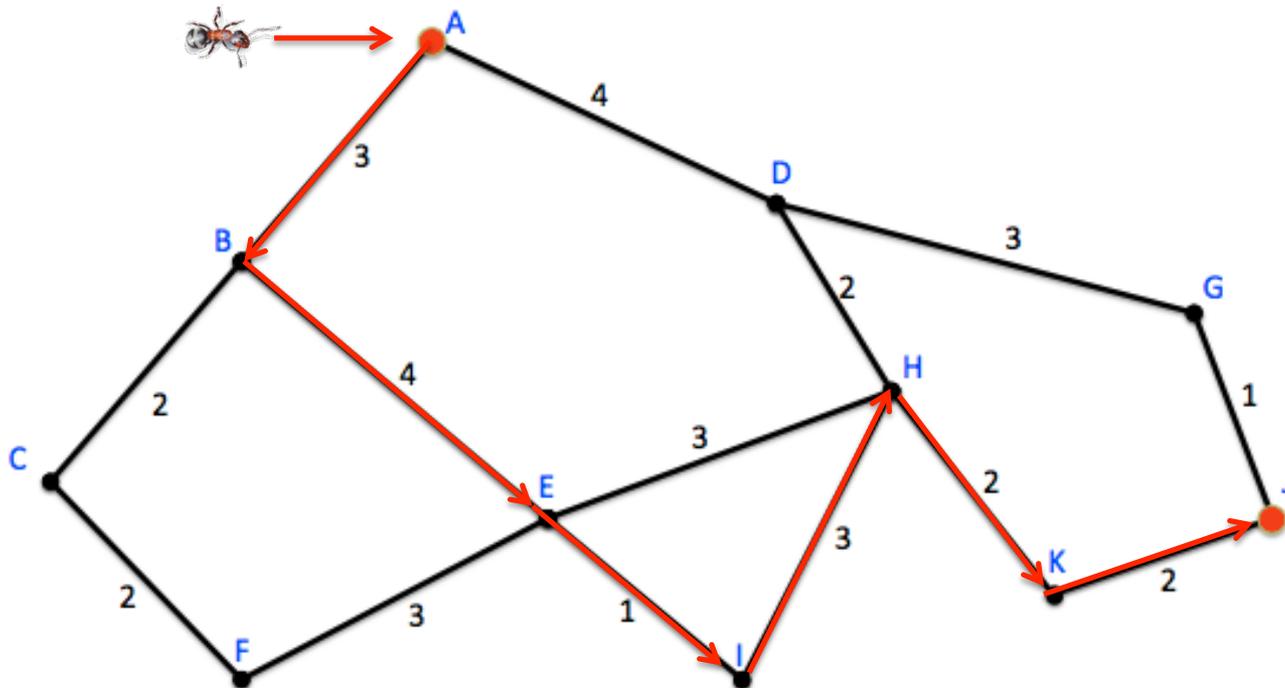
Schwarmintelligenz Der Ablauf als Pseudocode

- > Aufstellen des Graphen
- > Aufstellen einer Wahrscheinlichkeitstabelle für jeden Knoten
- > Aufstellen der mathematischen Formeln
- > Wiederhole:
 - Zufallswahl eines Ameisenweges
 - Anpassen der Wahrscheinlichkeiten der Tabelle jedes passierten Knotens
- > Bis Wegstrecke eindeutig



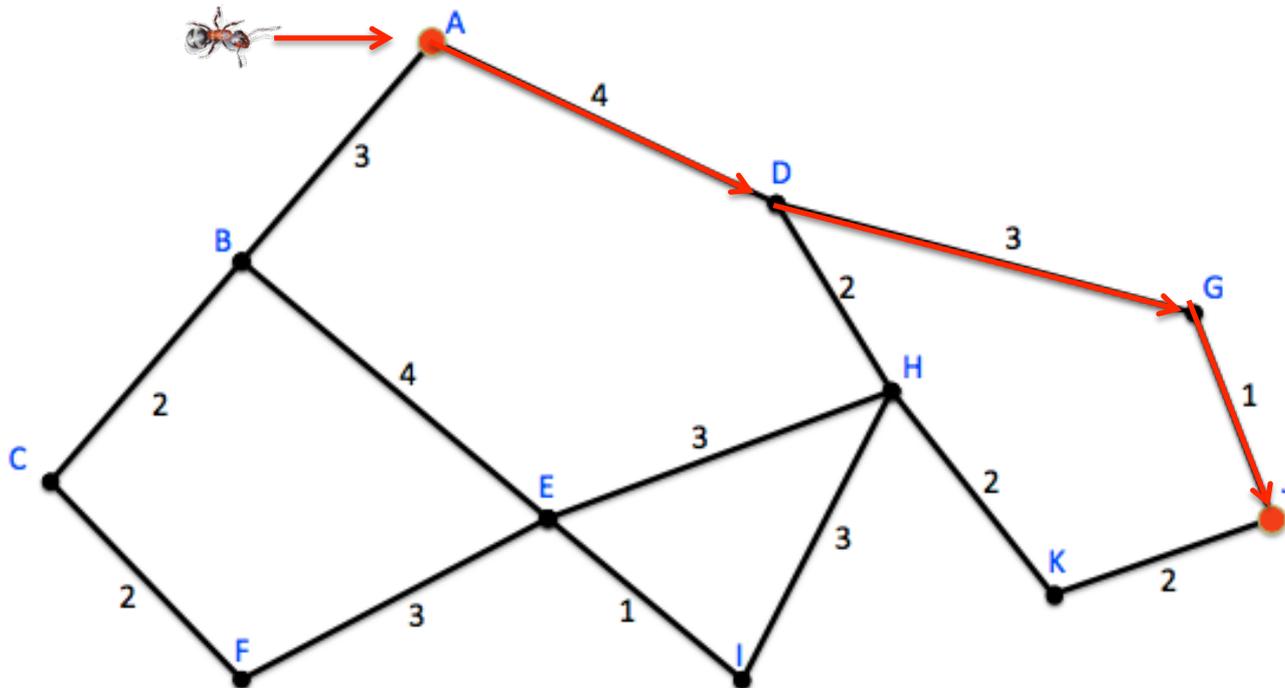
Schwarmintelligenz Der Ablauf als Pseudocode

- > Aufstellen des Graphen
- > Aufstellen einer Wahrscheinlichkeitstabelle für jeden Knoten
- > Aufstellen der mathematischen Formeln
- > Wiederhole:
 - Zufallswahl eines Ameisenweges
 - Anpassen der Wahrscheinlichkeiten der Tabelle jedes passierten Knotens
- > Bis Wegstrecke eindeutig

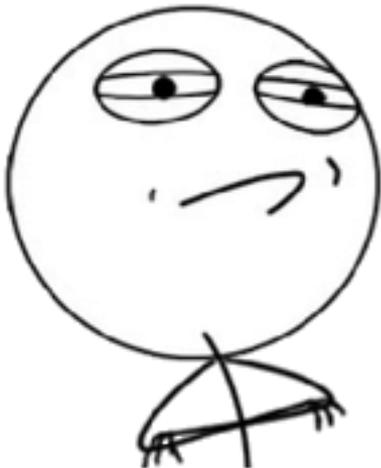


Schwarmintelligenz Der Ablauf als Pseudocode

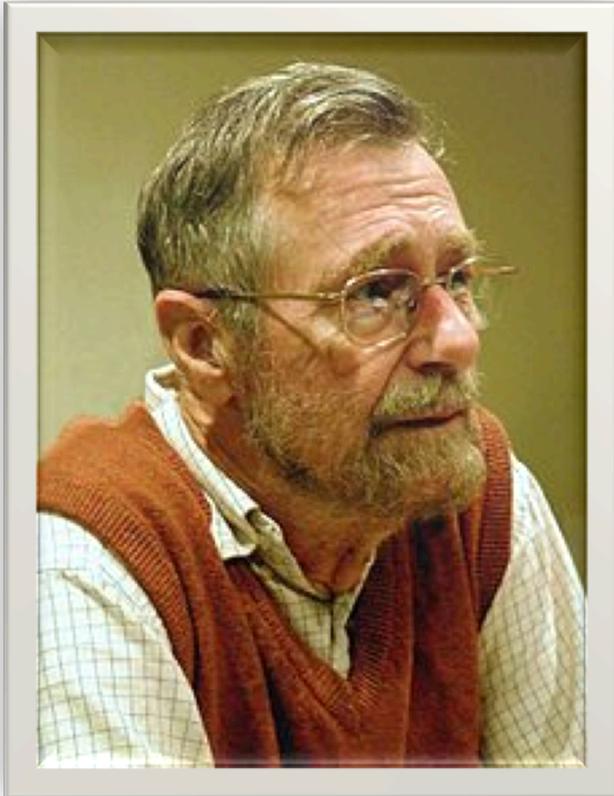
- > Aufstellen des Graphen
- > Aufstellen einer Wahrscheinlichkeitstabelle für jeden Knoten
- > Aufstellen der mathematischen Formeln
- > Wiederhole:
 - Zufallswahl eines Ameisenweges
 - Anpassen der Wahrscheinlichkeiten der Tabelle jedes passierten Knotens
- > Bis Wegstrecke eindeutig



Mein Dijkstra
schafft das auch!



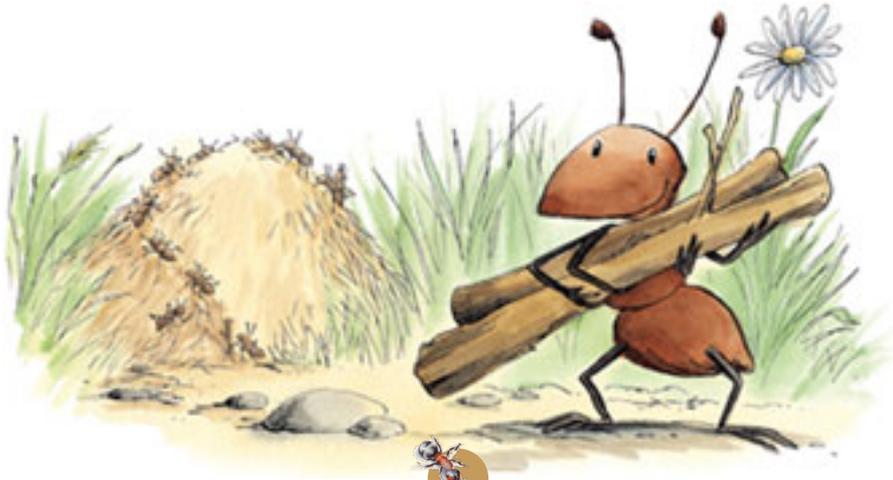
Exkurs Wer ist Dijkstra?



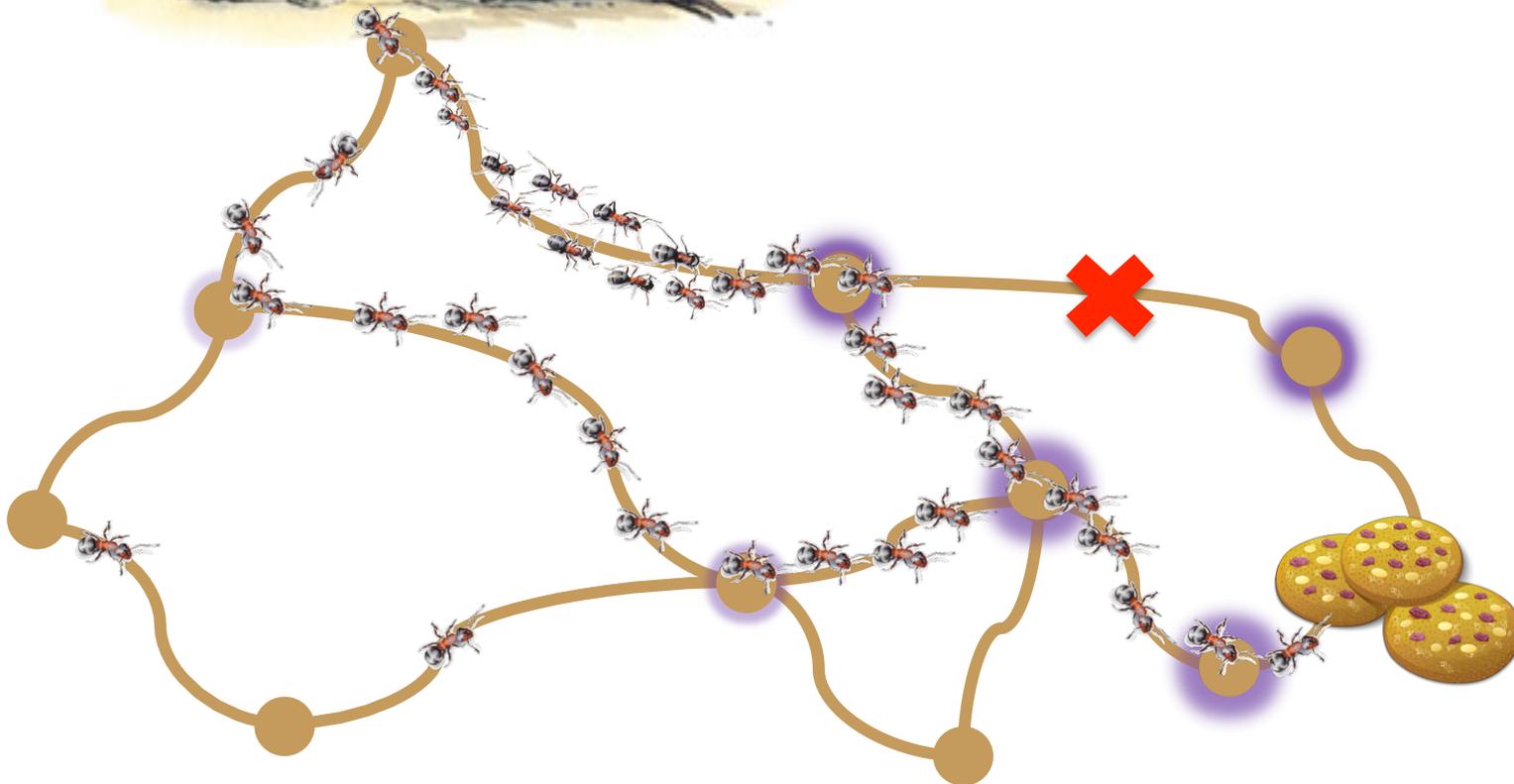
Edsger Wybe Dijkstra
1930 - 2002

- > Niederländischer Informatiker
- > Wegbereiter der strukturierten Programmierung
- > Erhielt 1972 den Turing Award
- > Meist in Verbindung mit dem Dijkstra-Algorithmus zur Berechnung des kürzesten Pfades innerhalb eines Graphen.

Schwarmintelligenz Hohe Dynamik durch Pheromone



Durch die Pheromon-Verteilung kann dynamisch auf die nächstbeste Route gewechselt werden.



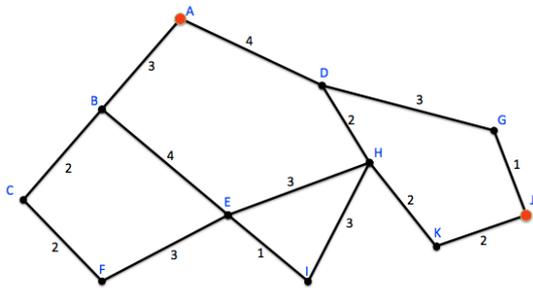
Schwarmintelligenz Statisch vs. dynamisch

Wann ist es ratsam die Ameisen loszulassen?

Statisch

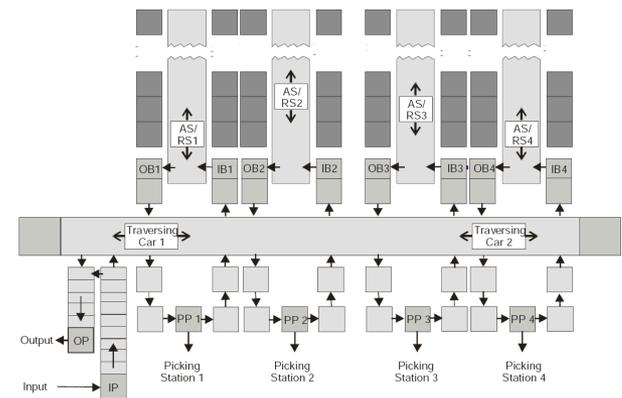
Halbdynamisch

Dynamisch



- > Fixer Graph, der nicht mehr verändert wird.
- > Schnelle optimale Berechnung durch statische Algorithmen wie Dijkstra

- > Graph ändert sich von Zeit zu Zeit.
- > Entscheidung ob jedes Mal Berechnung durch Dijkstra oder dynamisch



- > Graph in stetiger dynamischer Veränderung
- > Ständig können neue Zielpunkte oder Wegänderungen entstehen

Agenda

- > Schwarmintelligenz (25')
 - Was ist das?
 - Kleines Beispiel
 - Berechnung & Unberechenbarkeit
 - Ziele und Ausblicke

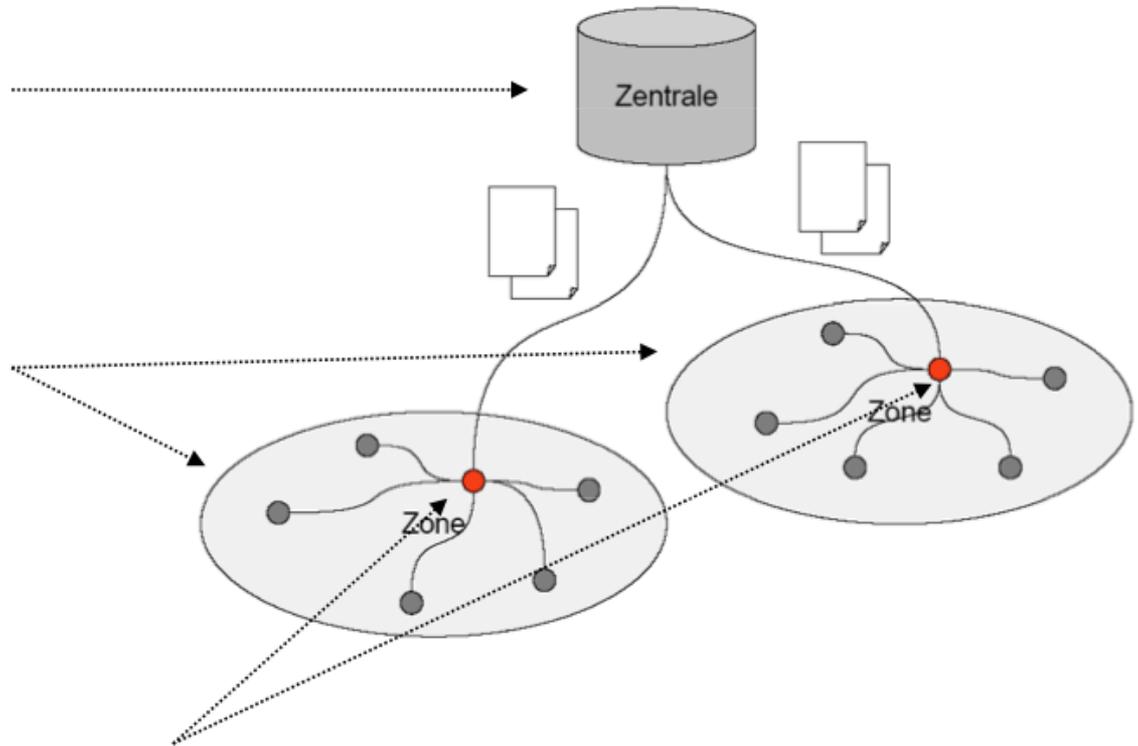
- > Ausgangspunkt (5')

- > Softwareentwicklung (30')

Schwarmintelligenz Ausblick: Navigation

Hier läuft der Ameisenalgorithmus

Die Pheromone jeder Zone werden an alle Fahrzeuge verteilt, die in dieser Zone fahren.



Nur wenige Fahrzeuge fragen direkt bei der Zentrale nach.

®Iwanowski

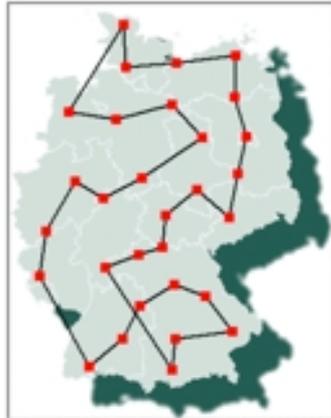
Schwarmintelligenz Ausblick: Spiele

- > Dynamische Reaktion auf Wegsperrungen
- > Unattraktive Wege durch ‚starken Beschuss‘
- > Ausreißer machen Spiellauf interessanter und individueller
- > Schwierigkeitsgrad durch simulierte Ameisen anpassbar

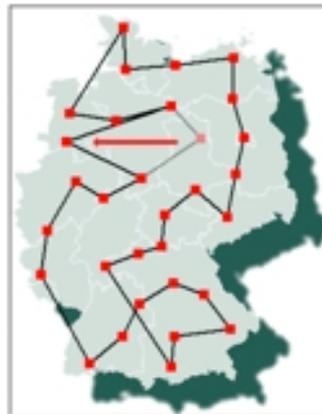


Desktop Tower Defense

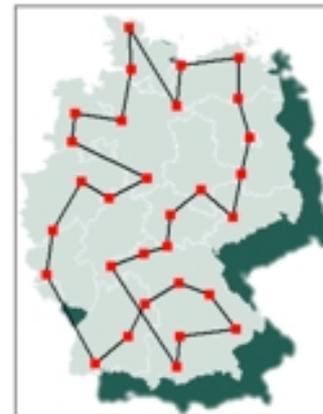
Schwarmintelligenz Ausblick: Travelling Salesman Problem



Nach kurzer Zeit
ist ein guter Weg
gefunden



Danach wird ein
Knoten verschoben



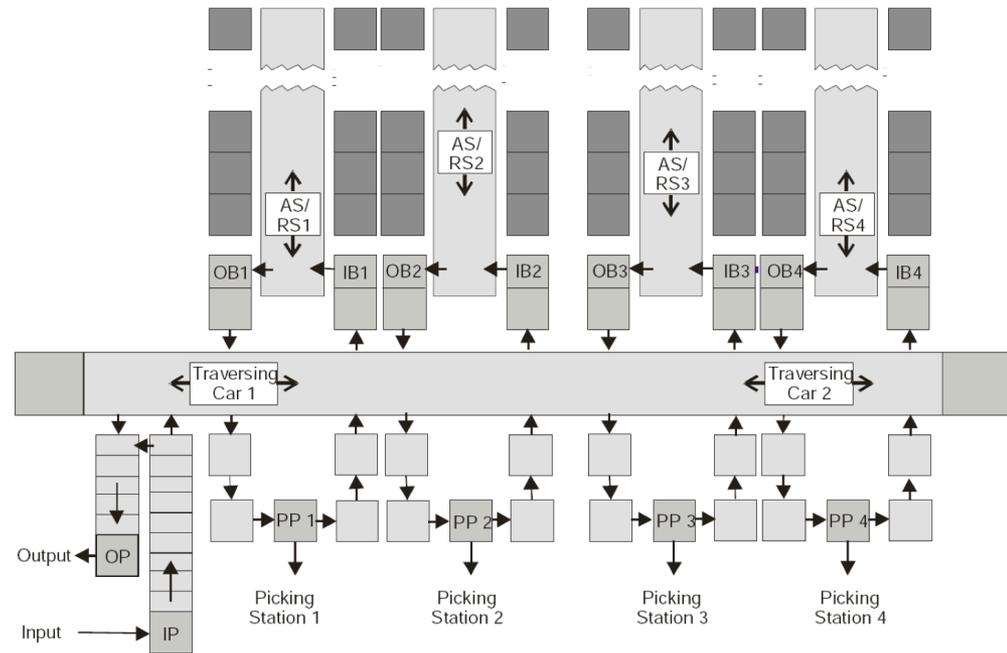
Kurz darauf hat sich
der Weg wieder
optimiert

Beispiel in der Logistik:

Anzahl der benötigten Fahrzeuge, als auch die jeweilige Wegstrecke soll optimiert werden. Da 2 Zielkriterien gegeben sind, werden auch 2 Ameisenkolonien erzeugt: Die erste Kolonie versucht einen guten Tourenplan zu finden, die andere Kolonie versucht die Anzahl der Fahrzeuge zu minimieren. Hat einer der Algorithmen eine besser Lösung gefunden, initialisiert er den anderen Algorithmus neu.

Schwarmintelligenz Ausblick: Logistik

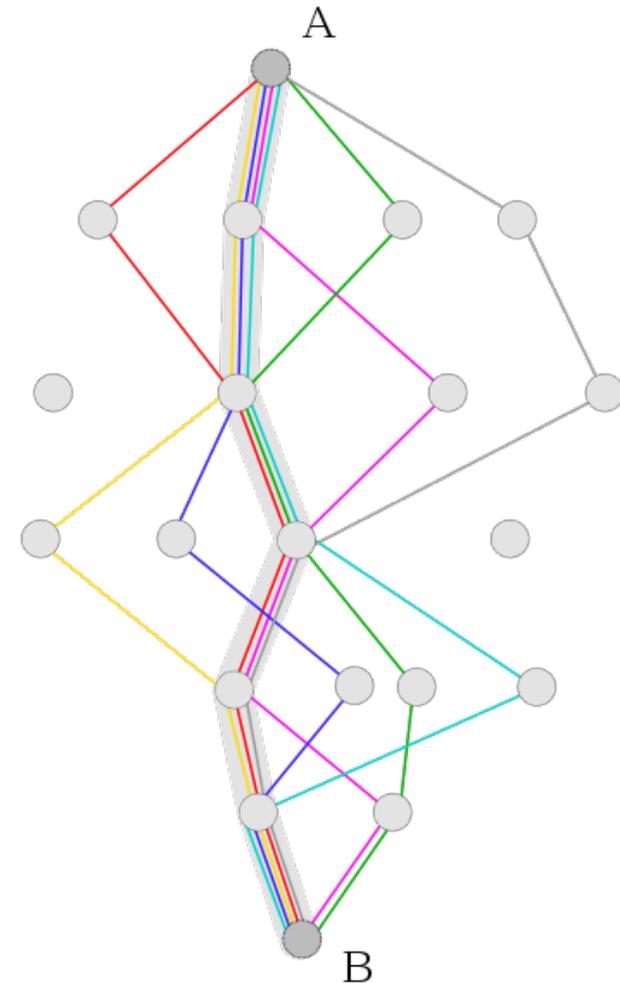
- > Im Warenhaus soll der kürzeste Weg durch das Lager gefunden werden.
- > Automatisierte Regalroboter bewegen sich zu Verpackungsstationen.
- > Unterwegs könnte dieser aber auch von einer weitem Station angefordert werden.
- > Dynamische Organisation wichtig für Zeit und Geld



HOMPEL, Michael ten; SCHMIDT, Thorsten: *Warehouse Management*. Springer, 2007

Schwarmintelligenz Ausblick: Algorithmische Optimierung

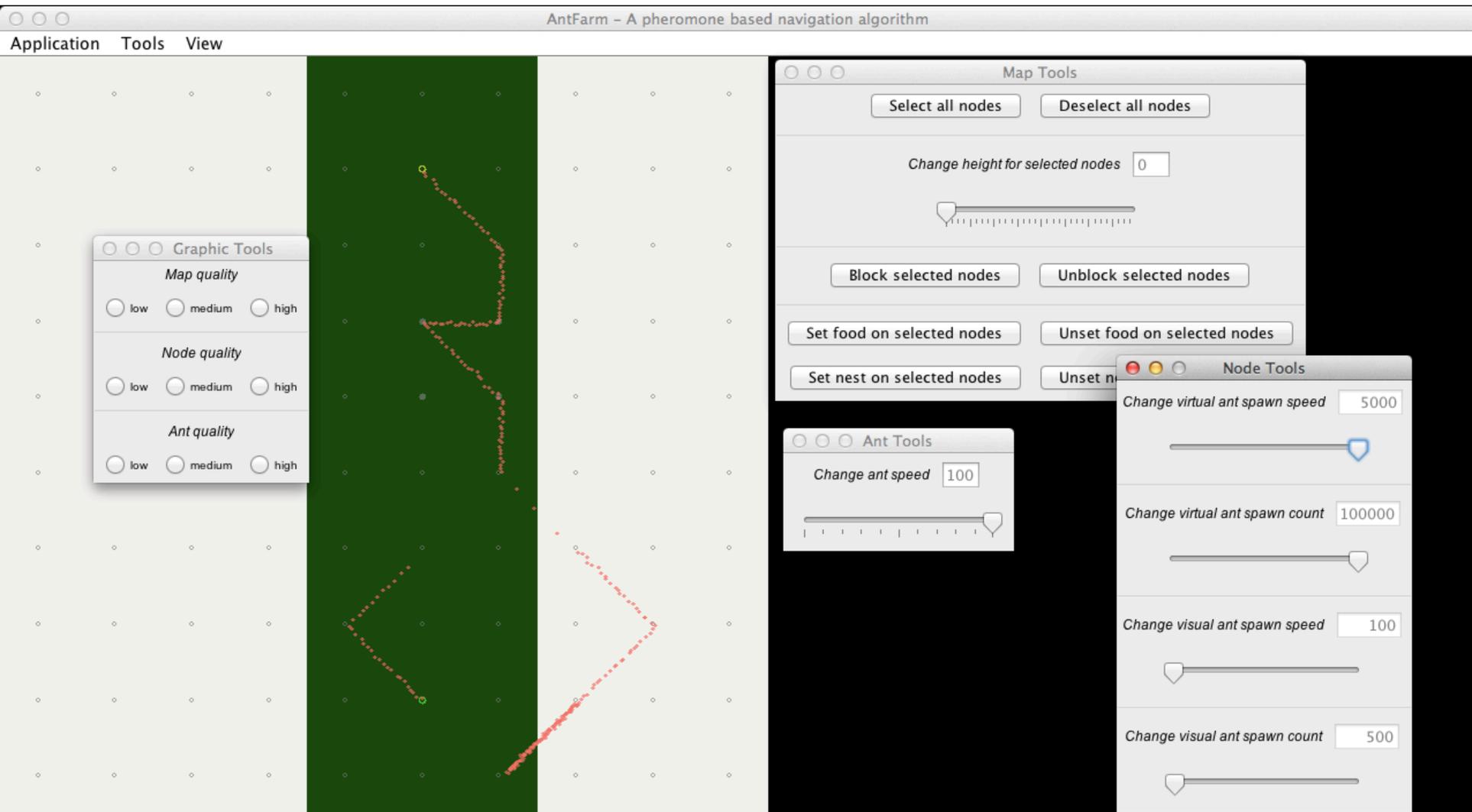
- > Busrouten, Müllabfuhr, Post- und Auslieferungsrouten.
- > Maschinenbelegungsproblem
 - > Minimierung der Transportzeit bei räumlich weit auseinander liegenden Produktionsstätten
- > Routenoptimierung
- > Beschickung von Lackieranlagen:
 - > Losbildung zur Minimierung der Farbwechsel
- > Fertigungssteuerung:
 - > Losbildung zur Minimierung der Rüstzeit und Einhaltung von Endterminen
- > Proteinfaltung:
 - > 20 Aminosäuren werden zu Proteinen mit 100 Aminosäuren kombiniert $\rightarrow 20^{100}$, dies ergibt etwa 10^{130} verschiedene Proteine.
- > Telefonnetzwerk und Internet
- > Personaleinsatzplanung bei Fluggesellschaften
- > Staplerleitsysteme



Agenda

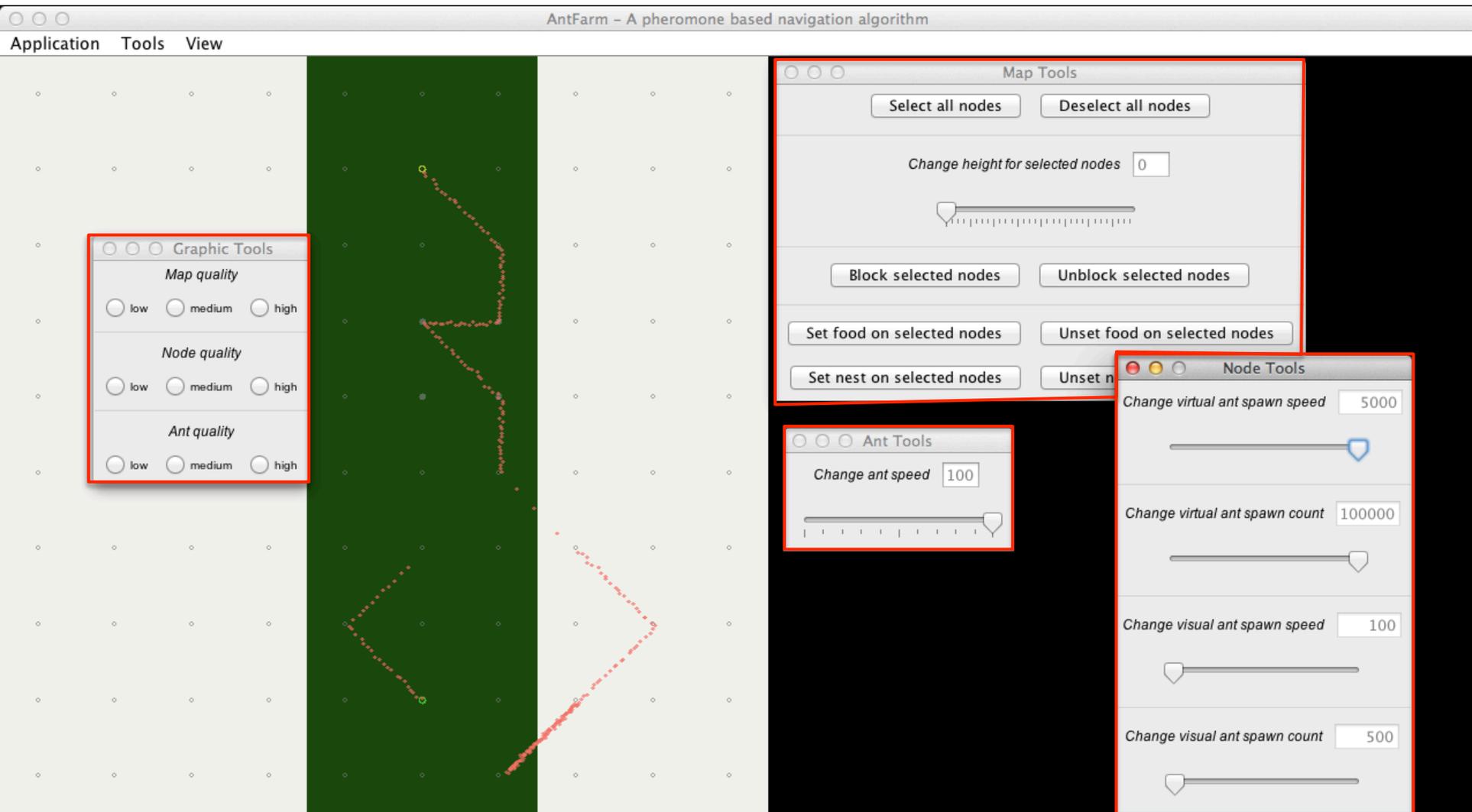
- > Schwarmintelligenz (25')
- > Ausgangspunkt (5')
 - Simulator v1.0
 - Schwachstellen
- > Softwareentwicklung (30')

Ausgangssituation Die Ameisen-Farm



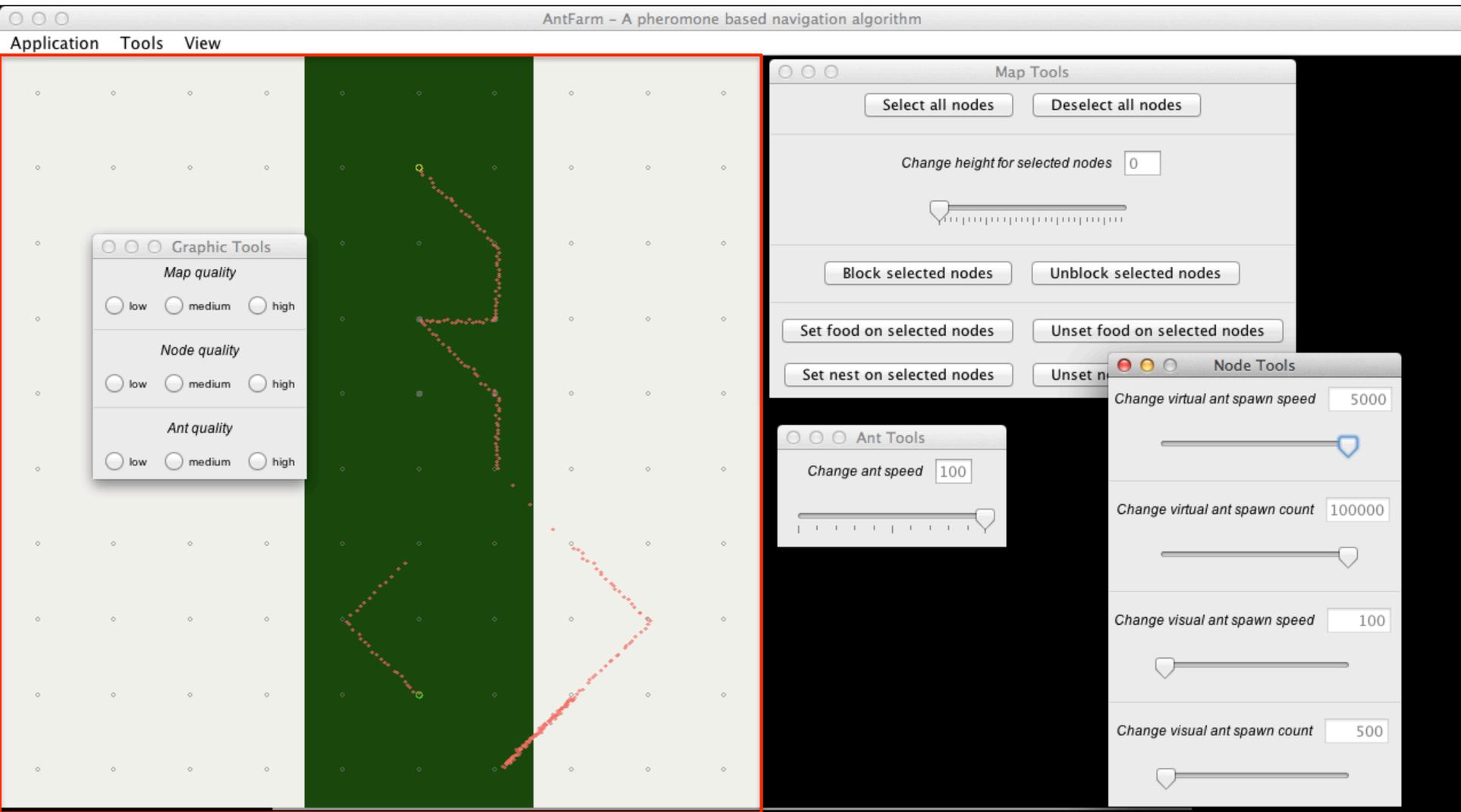
Ausgangssituation Die Ameisen-Farm

Verschiedene Frames erschweren einen Gesamtüberblick sowie das Handling



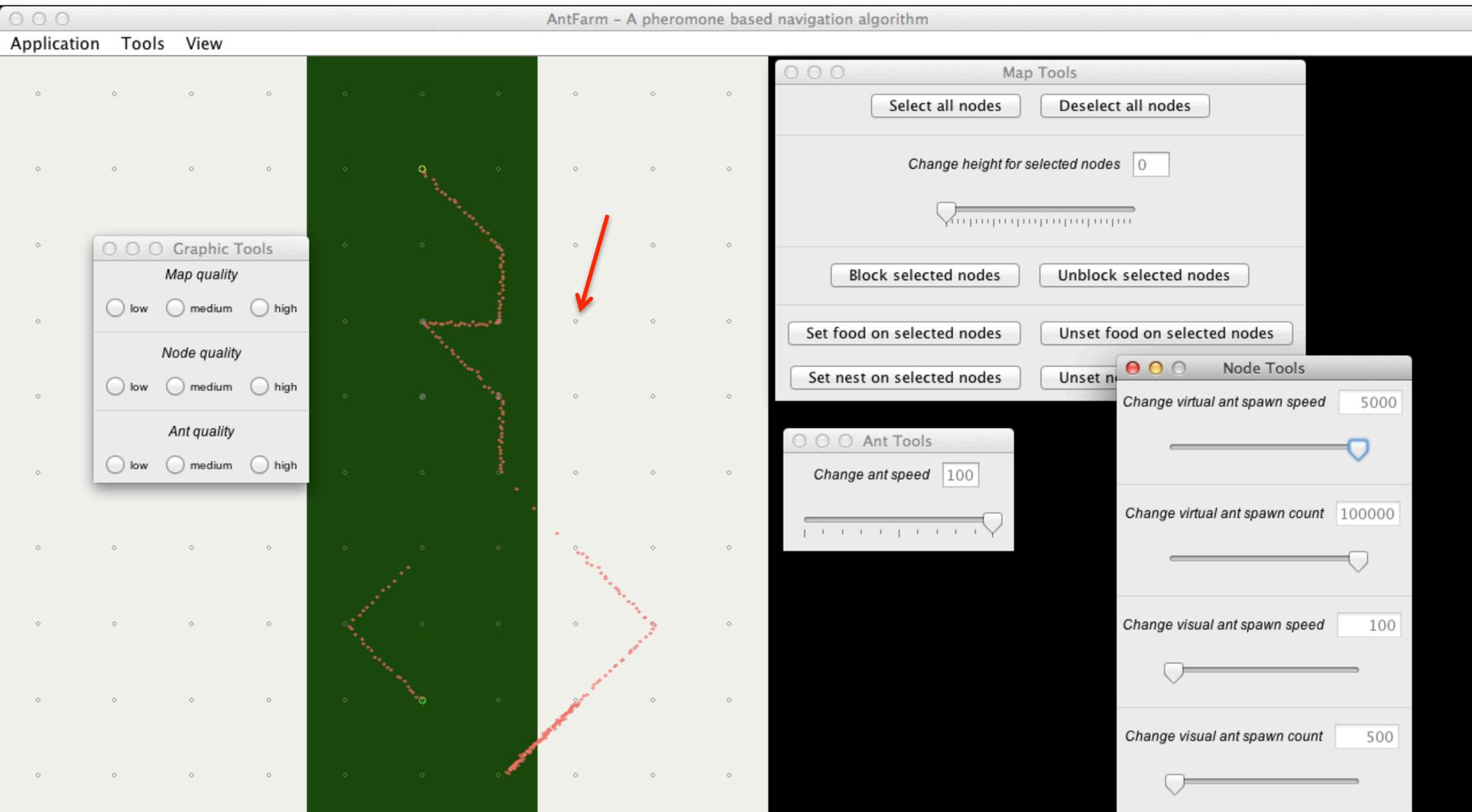
Ausgangssituation Die Ameisen-Farm

Auf der Karte sind die Höhen schwierig zu identifizieren



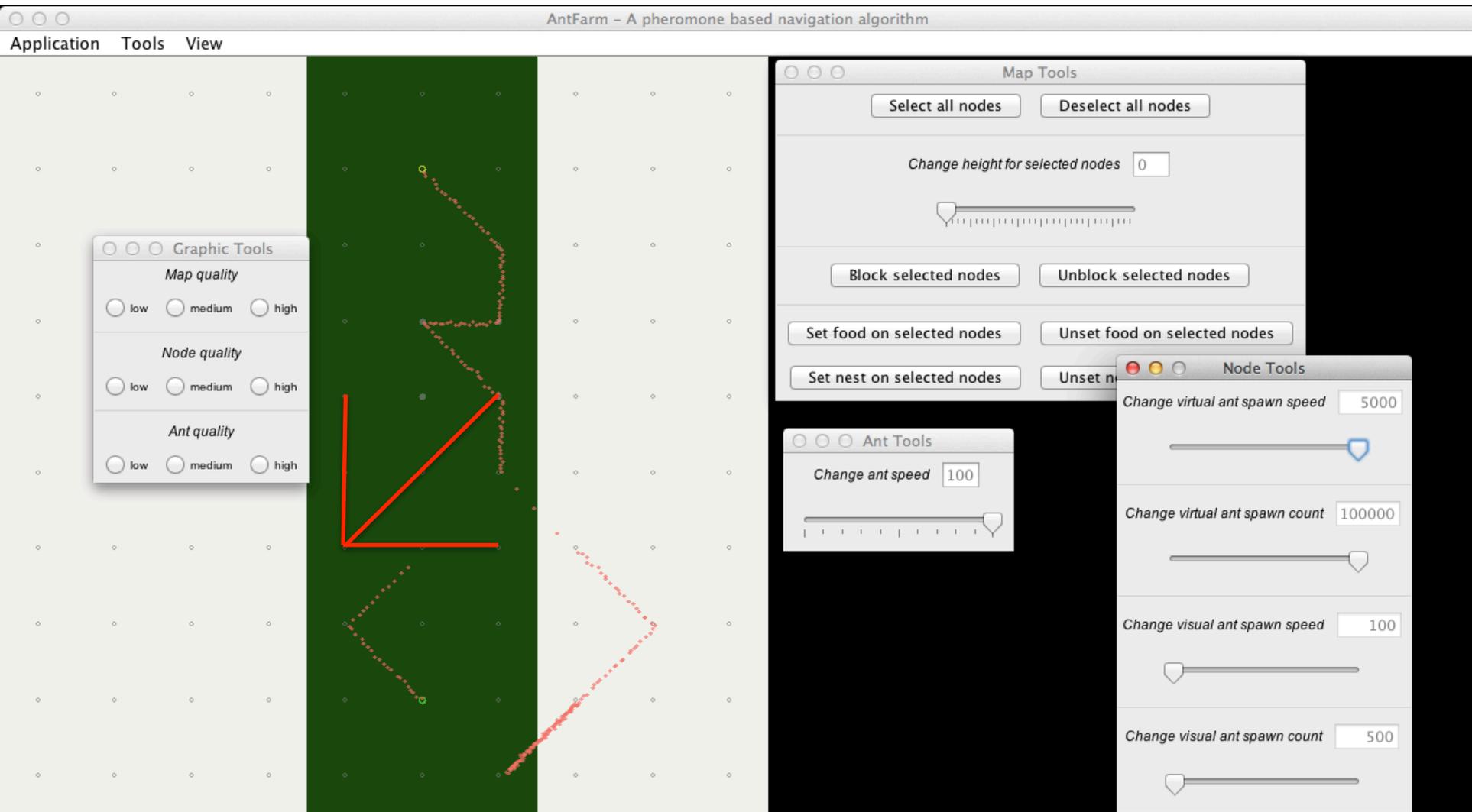
Ausgangssituation Die Ameisen-Farm

Knoten sind schwierig selektier- und anpassbar



Ausgangssituation Die Ameisen-Farm

Diagonale Strecken werden wie horizontale bzw. vertikale Strecken behandelt



Agenda

- > Schwarmintelligenz (25')
- > Ausgangspunkt (5')
- > Softwareentwicklung (30')
 - Problemstellung
 - Herangehensweise
 - Probleme, Lösungen, Lessons Learned
 - Lösung / Beispiel

Agenda

- > Schwarmintelligenz (25')
- > Ausgangspunkt (5')
- > Softwareentwicklung (30')
 - Problemstellung
 - Herangehensweise
 - Probleme, Lösungen, Lessons Learned
 - Lösung / Beispiel

Problemstellung

Ableitung der Arbeitspakete und Rahmenbedingungen

1) Laden der Karte

- > Im aktuellen Verzeichnis starten
- > Vorschau der Karte

2) Bedienung & GUI

- > Einstellungen an Feldern (z.B. Höhe) leichter gestalten
- > Blockieren klarer darstellen
- > Minimale Fenstergröße bestimmen
- > Sonstige Kleinigkeiten
- > **Einstellungs-Panel**
 - Teilung in Basic- und Advanced-Settings
 - Permanentes Anzeigen der Basic-Settings (Geschwindigkeit, Feldhöhe, Futter, Nest)
 - Optionales Anzeigen der Advanced-Settings (Spawn-Speed, Spawn-Count, Grafikeinstellungen)

3) Kernalgorithmus

- > Diagonalen-Länge anpassen

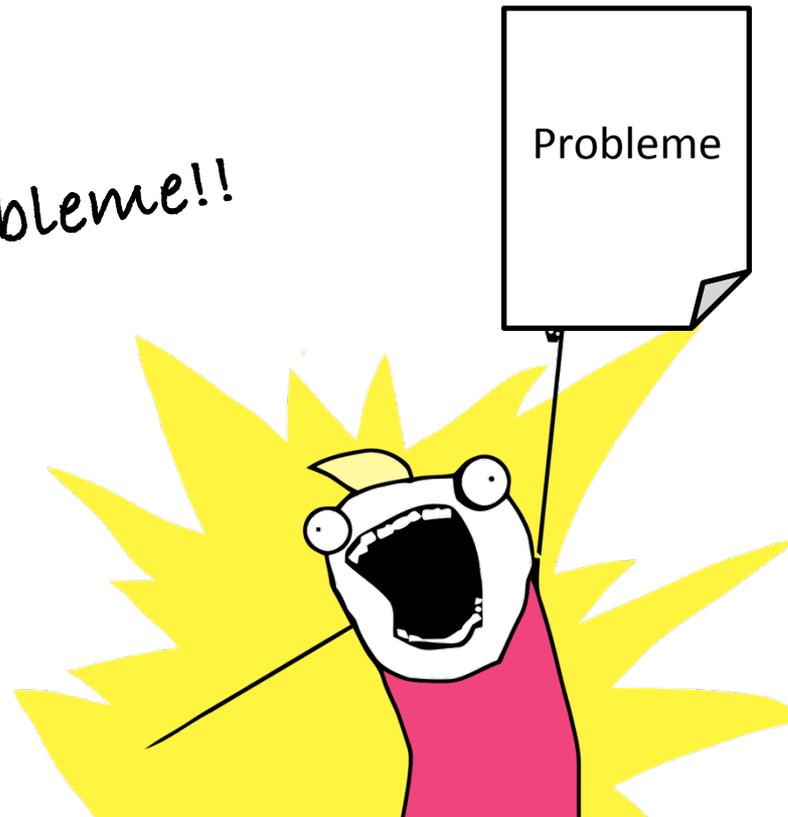
4) Zeit und Aufwand

- > 2 ECTS (= 60 Arbeitsstunden) pro Person
- > Abschluss bis zum Tag der offenen Tür (etwa 2 Monate)

Agenda

- > Schwarmintelligenz (25')
- > Ausgangspunkt (5')
- > Softwareentwicklung (30')
 - Problemstellung
 - **Herangehensweise**
 - Probleme, Lösungen, Lessons Learned
 - Lösung / Beispiel

Wir lösen alle Probleme!!

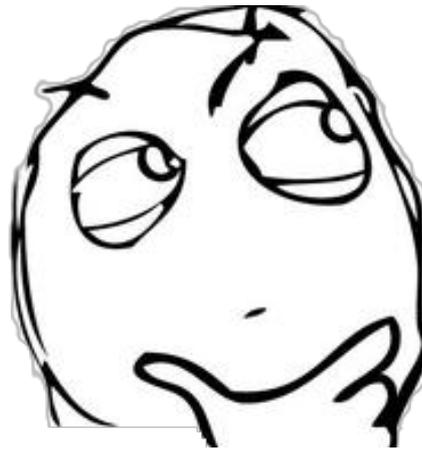


Herangehensweise Das Team



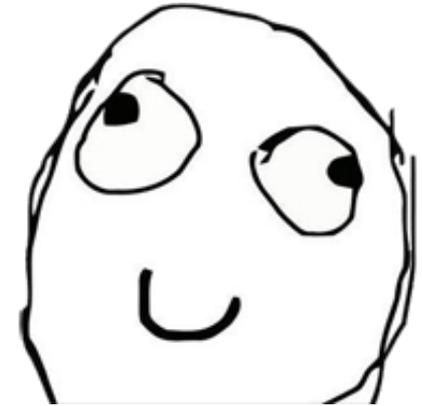
Christoph – Der Macher

- > Vorkenntnisse in Java
- > Internationale Projektmanagement-erfahrung



Lennart – Der Planer

- > Erfahrungen durch PTL, Nebenjob und Hobby
- > Projekterfahrung als Berater



Niko – Der Kreative

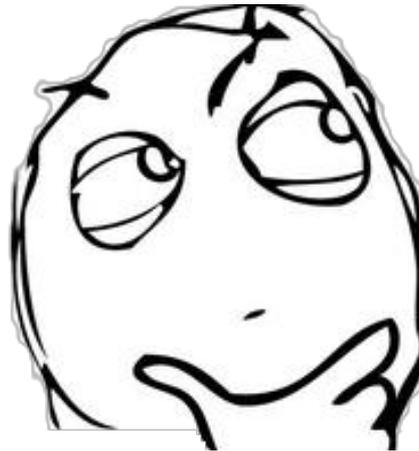
- > Erste Programmier-erfahrungen durch PS 1, PS 2 und C

Herangehensweise Das Team



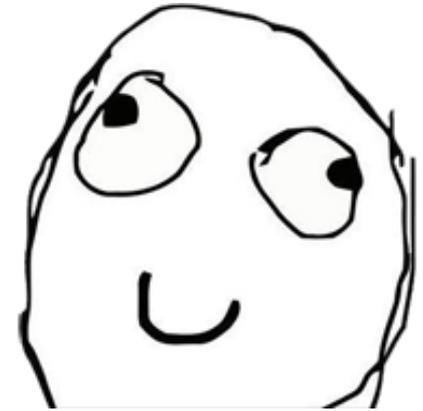
Christoph – Der Macher

- > Einbauen der erweiterten Einstellungen
- > Erstellen von Grafiken



Lennart – Der Planer

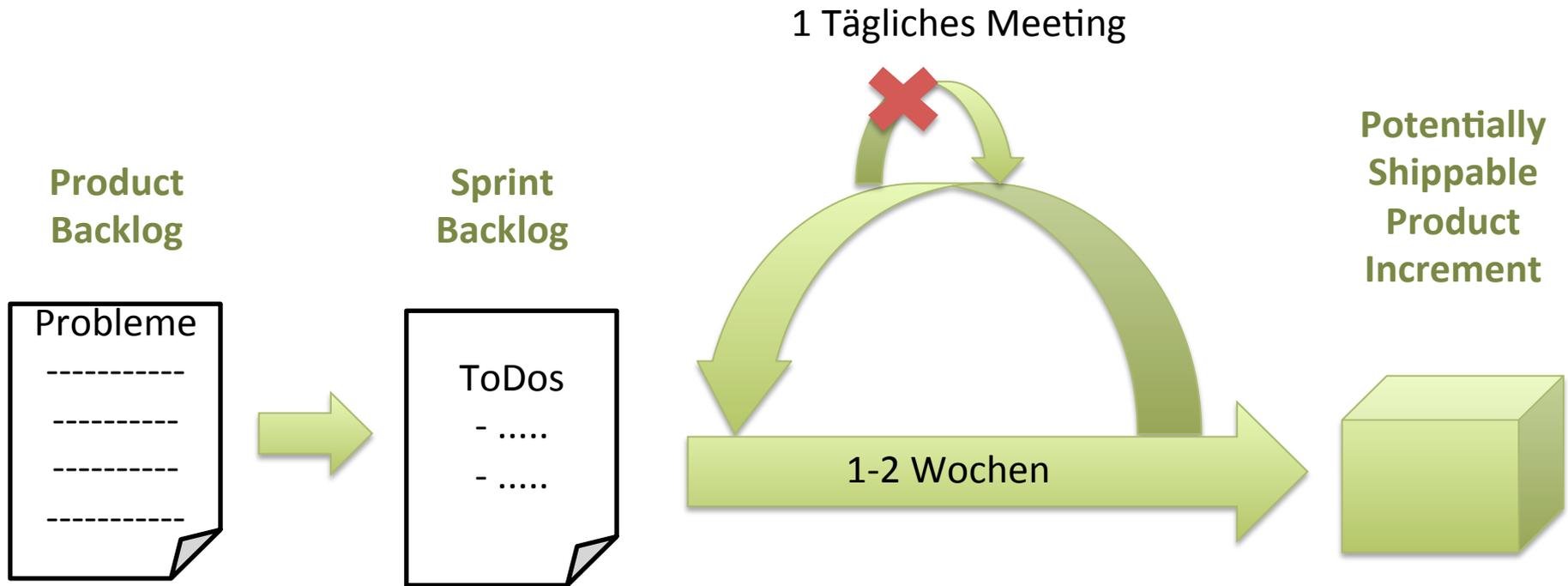
- > Struktur und Bedienung
- > Verwalten und Herstellen von Schnittstellen



Niko – Der Kreative

- > Anpassungen der Diagonalenlängen im Kernalgorithmus
- > Algorithmische Anpassung der Blockaden

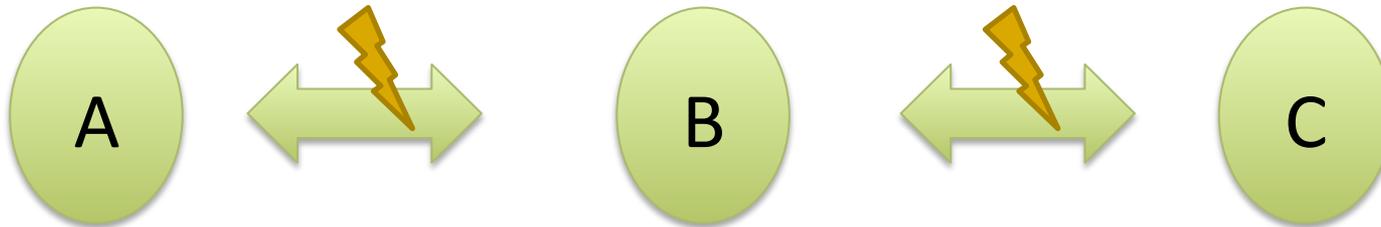
Herangehensweise Innere Organisation - Scrum



Agenda

- > Schwarmintelligenz (25')
- > Ausgangspunkt (5')
- > Softwareentwicklung (30')
 - Problemstellung
 - Herangehensweise
 - Probleme, Lösungen, Lessons Learned
 - Beispiel

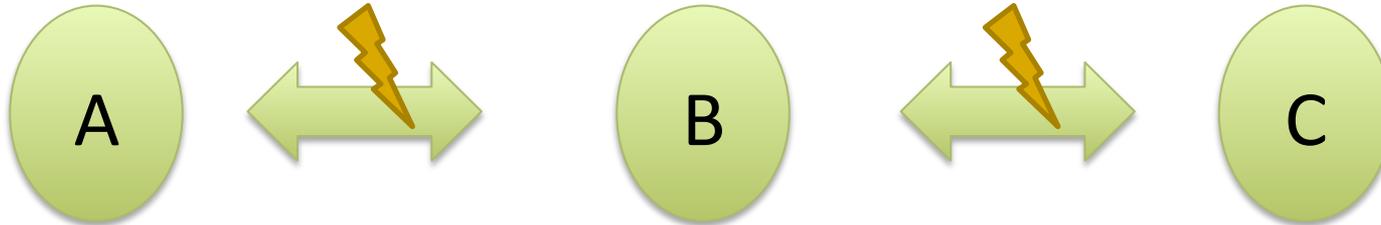
Problem: Verschiedene Fähigkeiten



Bei verschiedenen Fähigkeiten gibt es zwangsläufig Kommunikationsbedarf, aber ...

- > In der Regel bleiben E-Mails an Gruppen **unbeantwortet** (insbesondere ohne Projektleiter)

Lesson Learned: Auf Probleme eingehen



Bei verschiedenen Fähigkeiten gibt es zwangsläufig Kommunikationsbedarf, aber ...

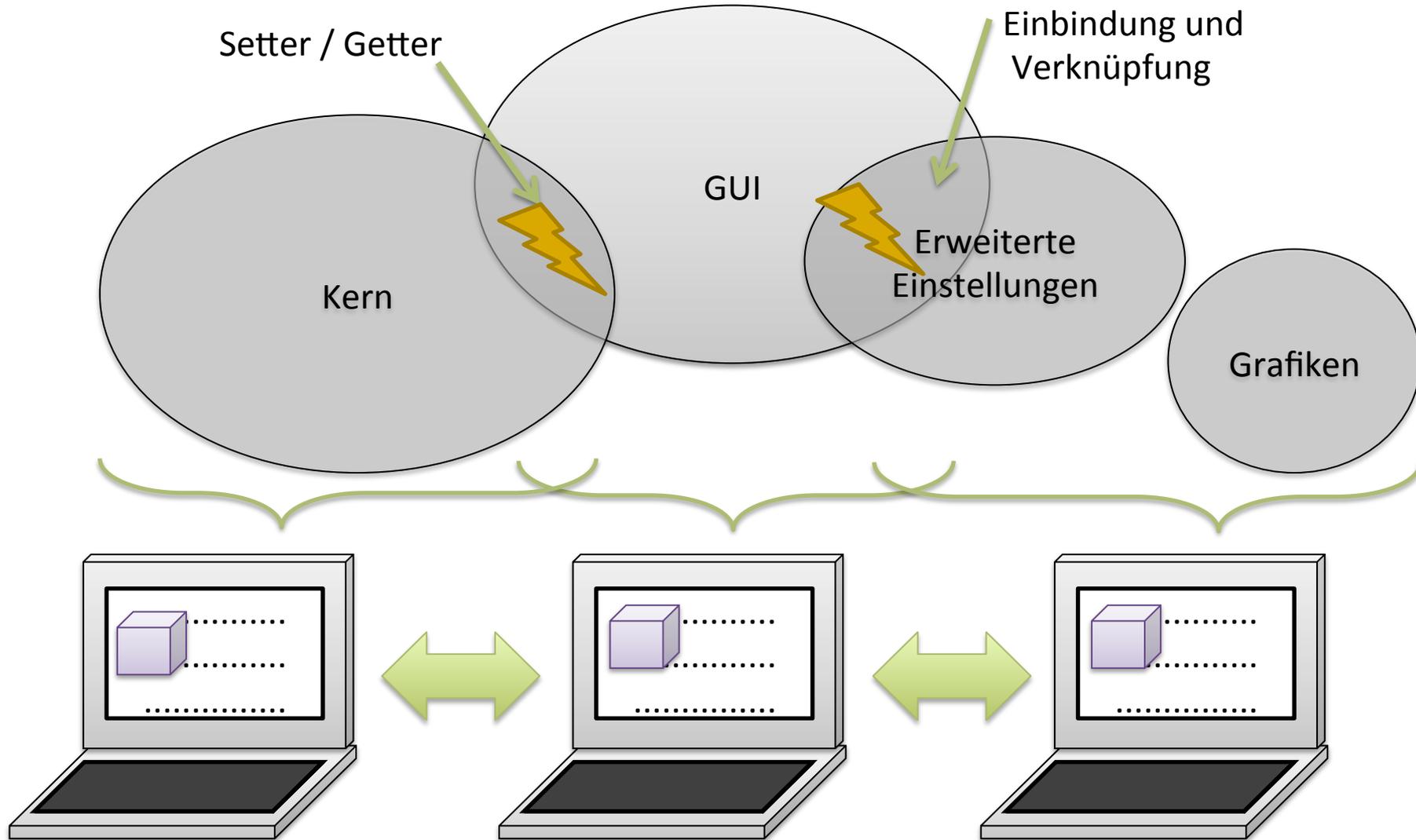
- > In der Regel bleiben E-Mails an Gruppen **unbeantwortet** (insbesondere ohne Projektleiter)

Daher:

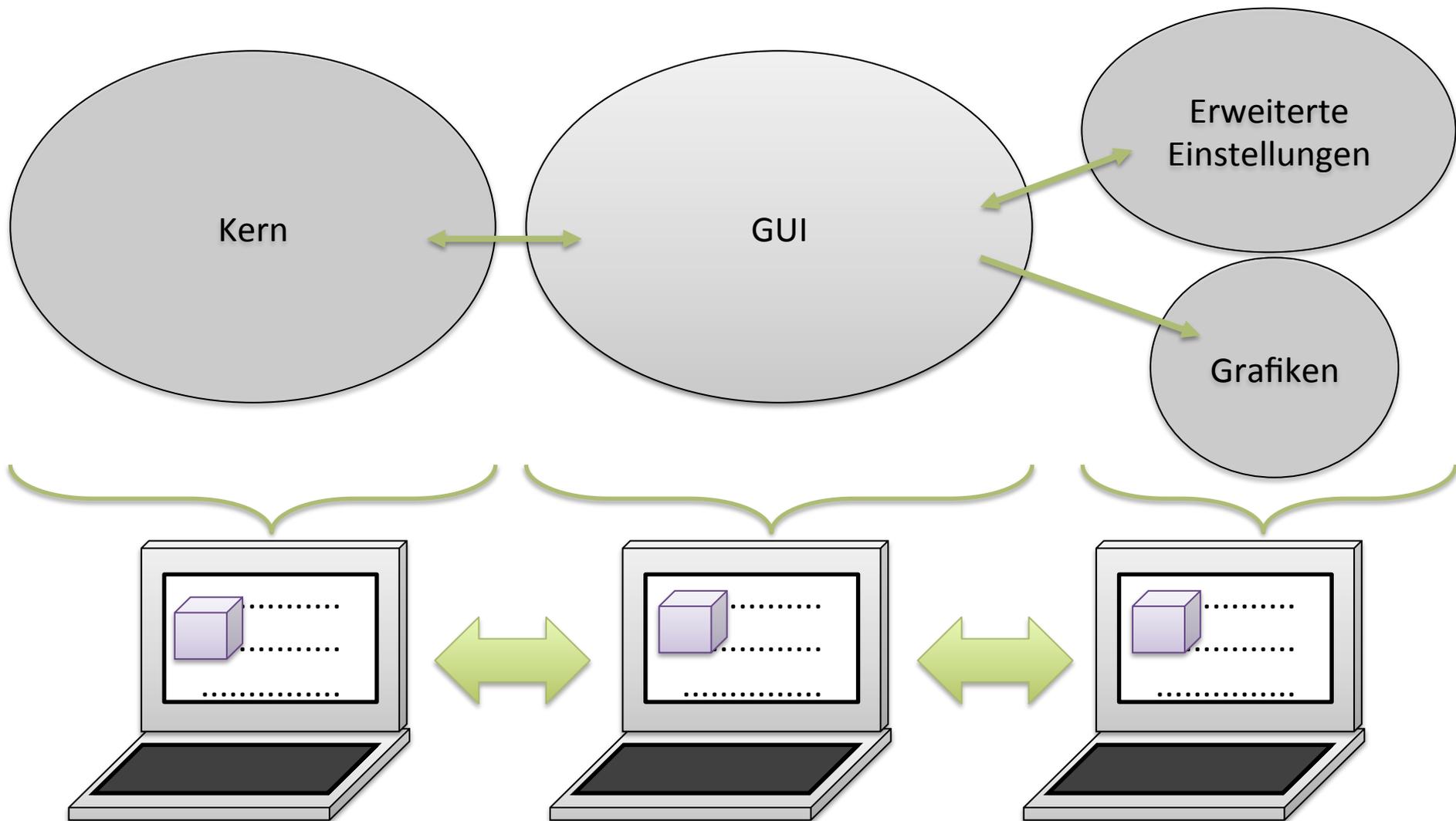
- > Hilfe nicht nur textuell suchen (besonders nicht an eine Gruppe)
- > Telefon, Skype, persönliches Gespräch und Teammeeting nutzen!

Offenes Ohr für ‚Hilferufe‘ haben!
Projektarbeit kann sonst ins Stocken geraten

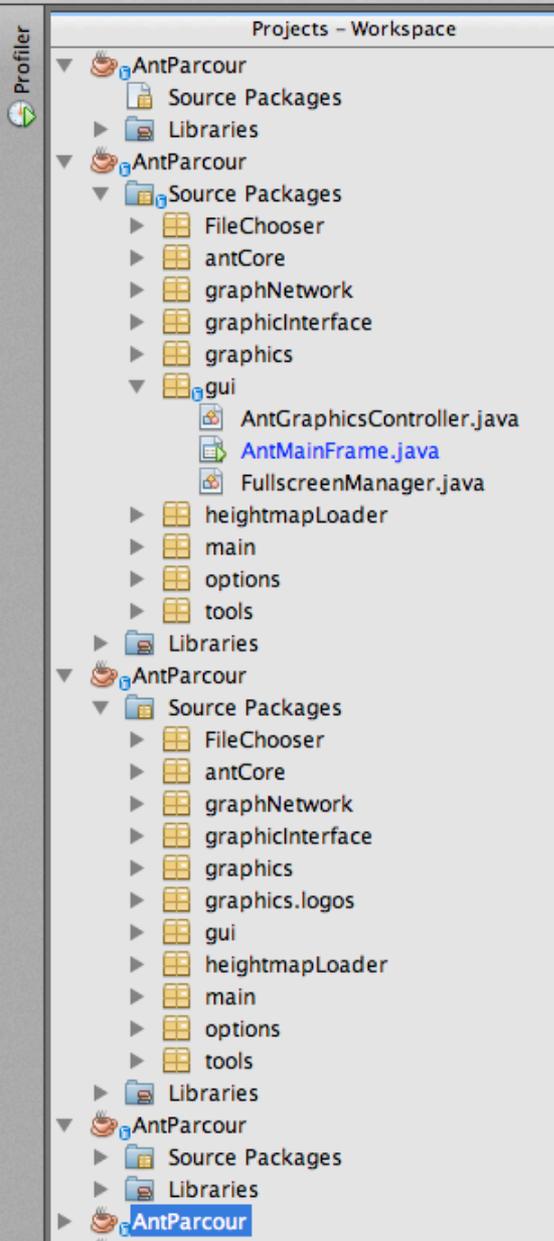
Problem: Arbeitsteilung und Synchronisation



Lesson Learned: Arbeitsteilung auf Klassen aufspalten



Problem: Repository & Backup



Auf einmal sind Features weg, weil man nicht auf der neusten Version gearbeitet hat. WAS TUN??

Lösung: Backup

Dropbox



- + Dynamisch in Workspace integrierbar
 - + Live-Synchronisation (nicht auf Nachfragen)
 - ‚Wirkliche‘ Versionierung nur in Abo-Version ab 140\$ p.a.
 - Kein Merging möglich, nur Konflikte mit Codeverdopplung
 - Kein Backup für den Ernstfall
-

Subversion

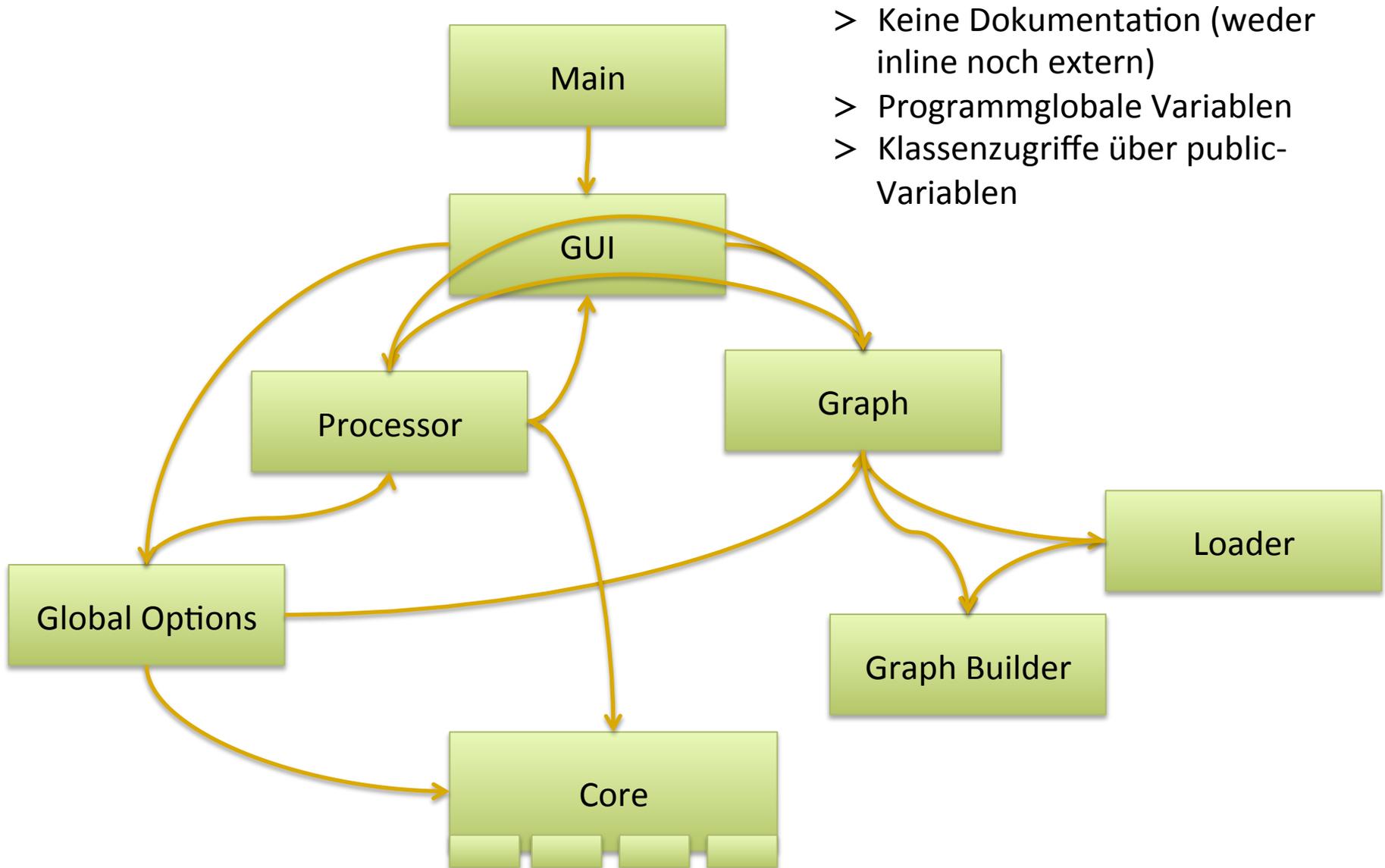


- + Versionierung
 - o Einbinden in IDE frickelig
 - Aufwand zur Konfliktbehebung relativ hoch
 - Synchronisation im src-Ordner schwierig
 - Kostenpflichtig
 - Kompletter Workspace-Backup in Konkurrenz zur Synchronisation
-

Regelmäßiges lokales Backup

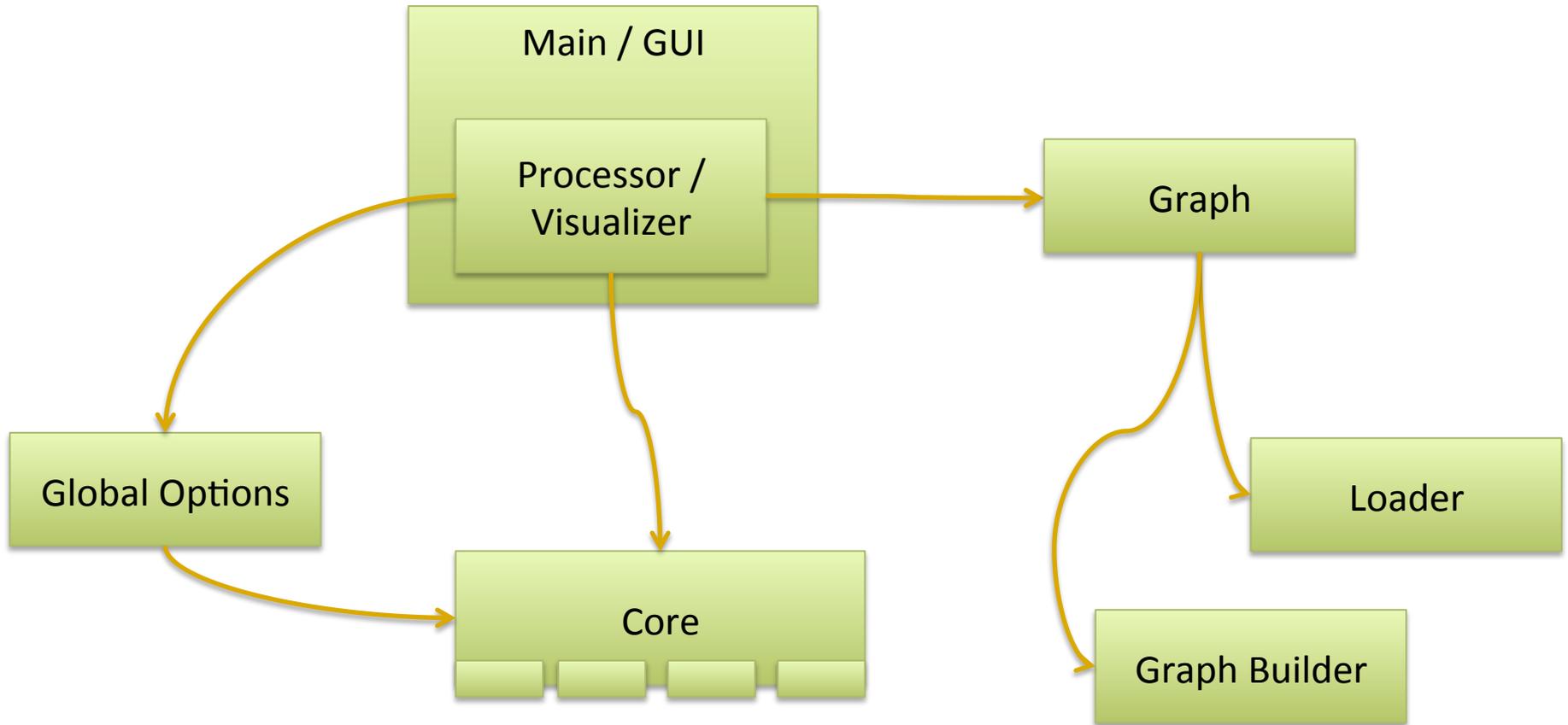
Sichert Projektinhalte vor Serverchaos, vergessenem Passwort, etc..

Problem: Orientierung im vorhandenen Code



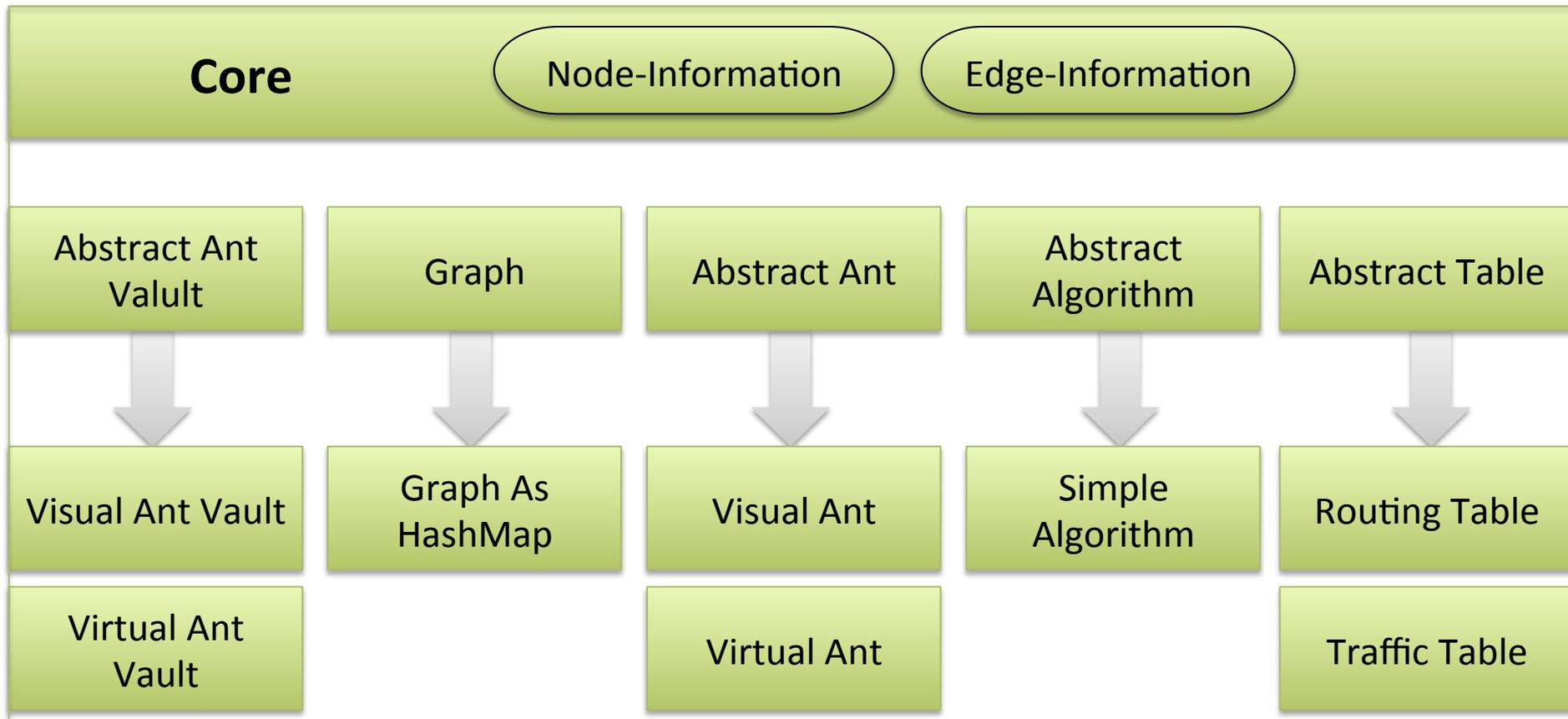
Lösung: Strukturierung des Codes

- > Dokumentation der Einstiegsklassen als JavaDoc
- > Ablauf und Übersicht vereinfacht



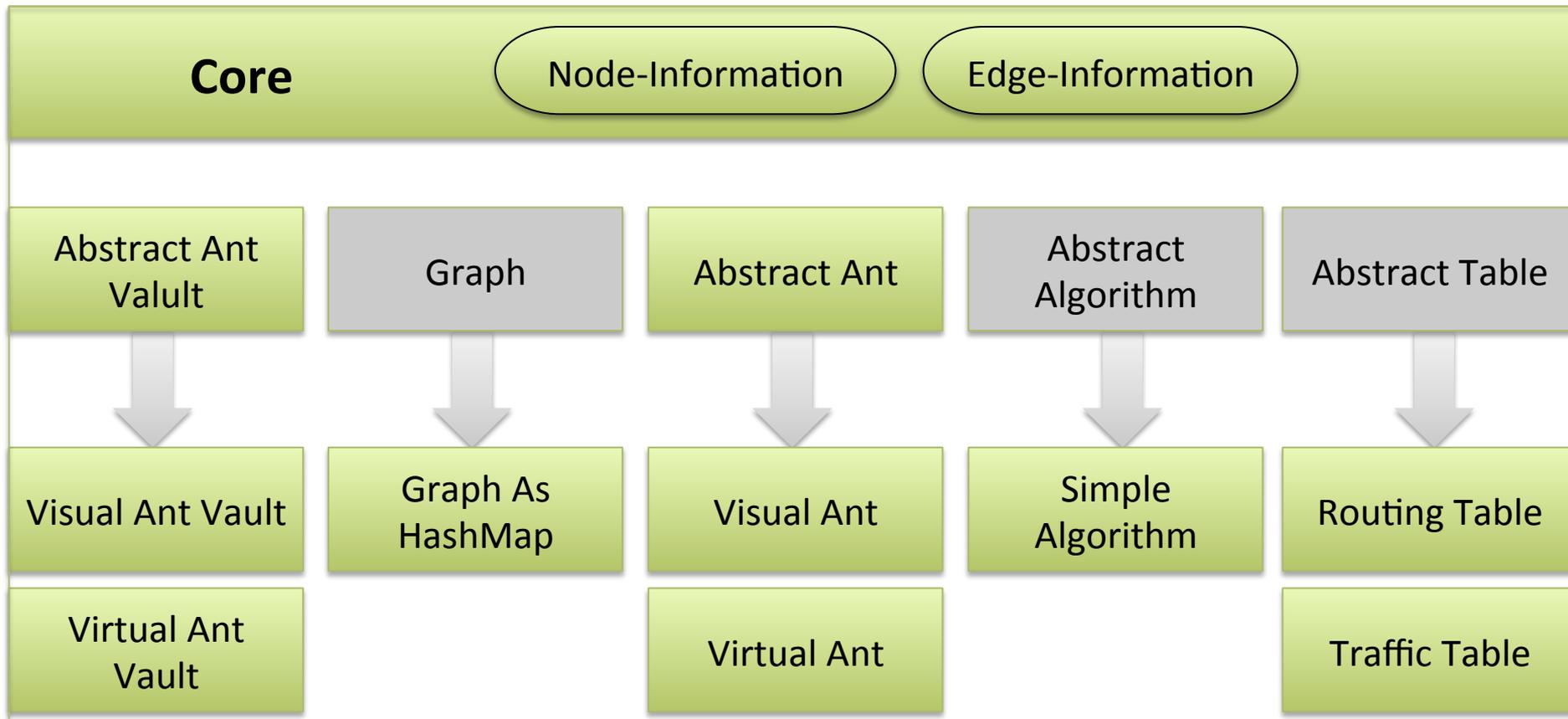
Problem: Zu viel Abstraktion macht das Einarbeiten schwer

Abstraktion ist elementarer Bestandteil der Programmierung.
Jedoch führt der übertriebene Einsatz auch schnell zu Unübersichtlichkeit.
Insbesondere Anfängern fällt der Einstieg dann schwer.



Lesson Learned: Abstraktion mit Sinn und Verstand

Abstraktion ist elementarer Bestandteil der Programmierung.
Jedoch führt der unüberlegte/übertriebene Einsatz auch schnell zu Unübersichtlichkeit.
Insbesondere Anfängern fällt der Einstieg dann schwer.



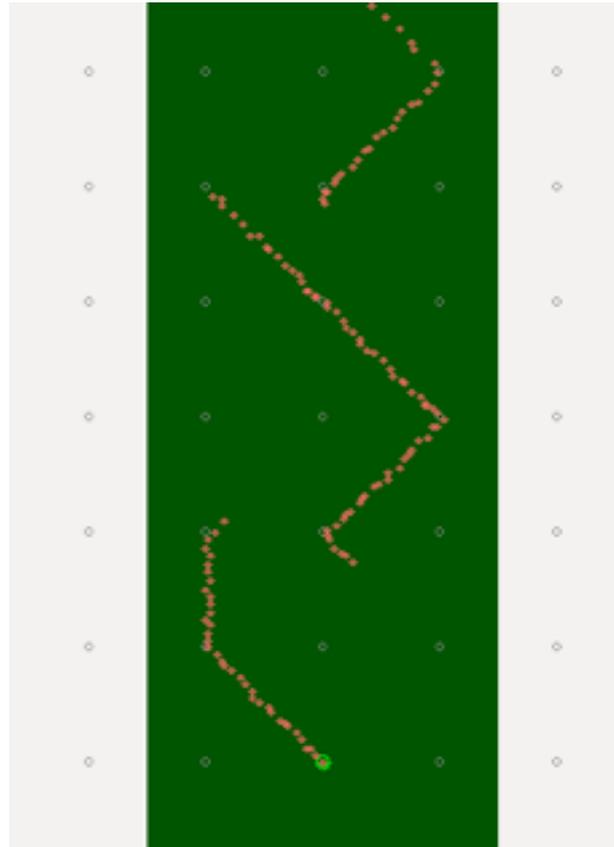
Lesson Learned: Gute Strukturen sind am Wichtigsten

- > Gute Struktur + Dokumentation
- > Zugriffe über public-Variablen in anderen Klassen sind böse!!
- > Programm-globale Variablen verschleiern Einstellungen einer Klasse.
- > Zu viel Abstraktion lässt Code unleserlich werden



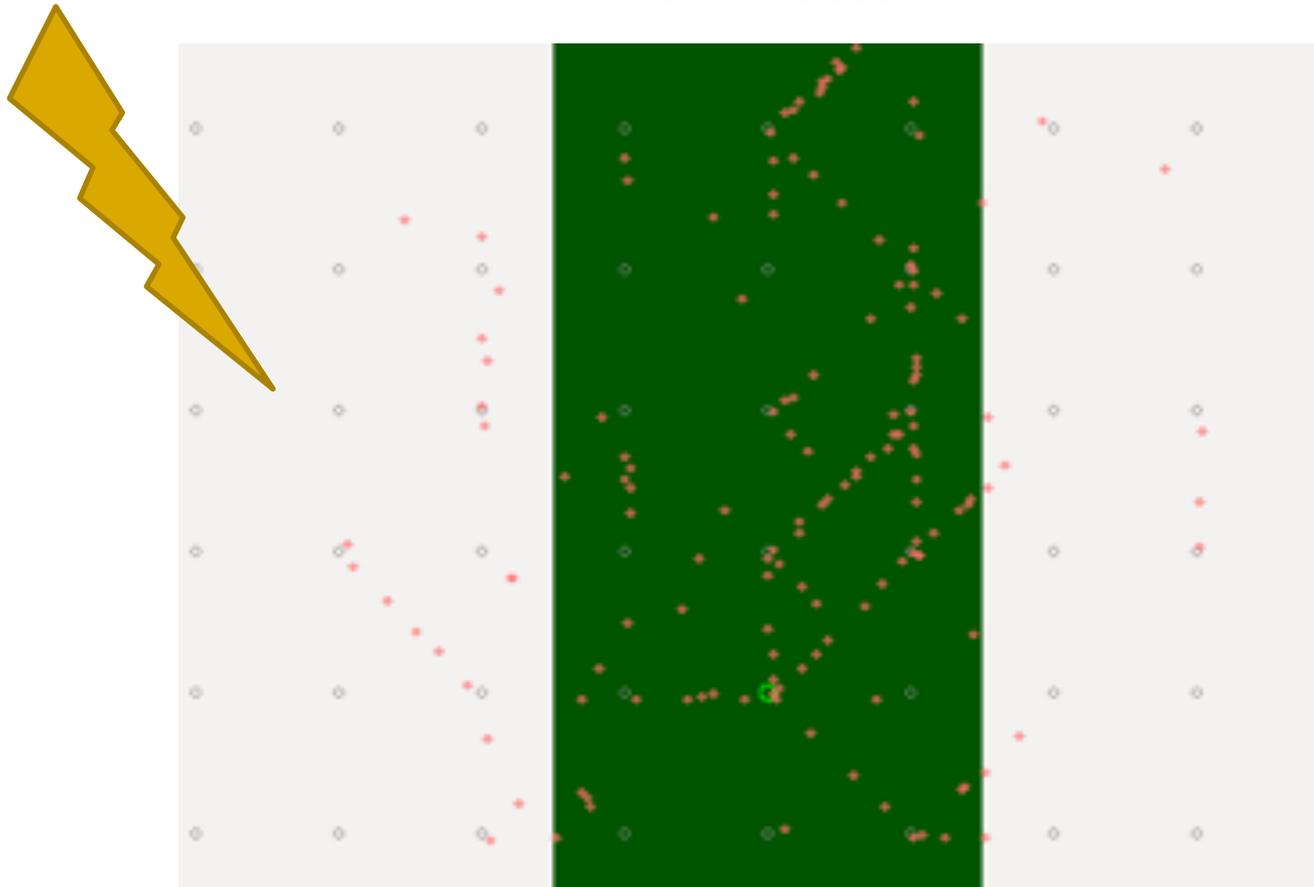
Problem: Code-Differenzen

So sollte es aussehen ...

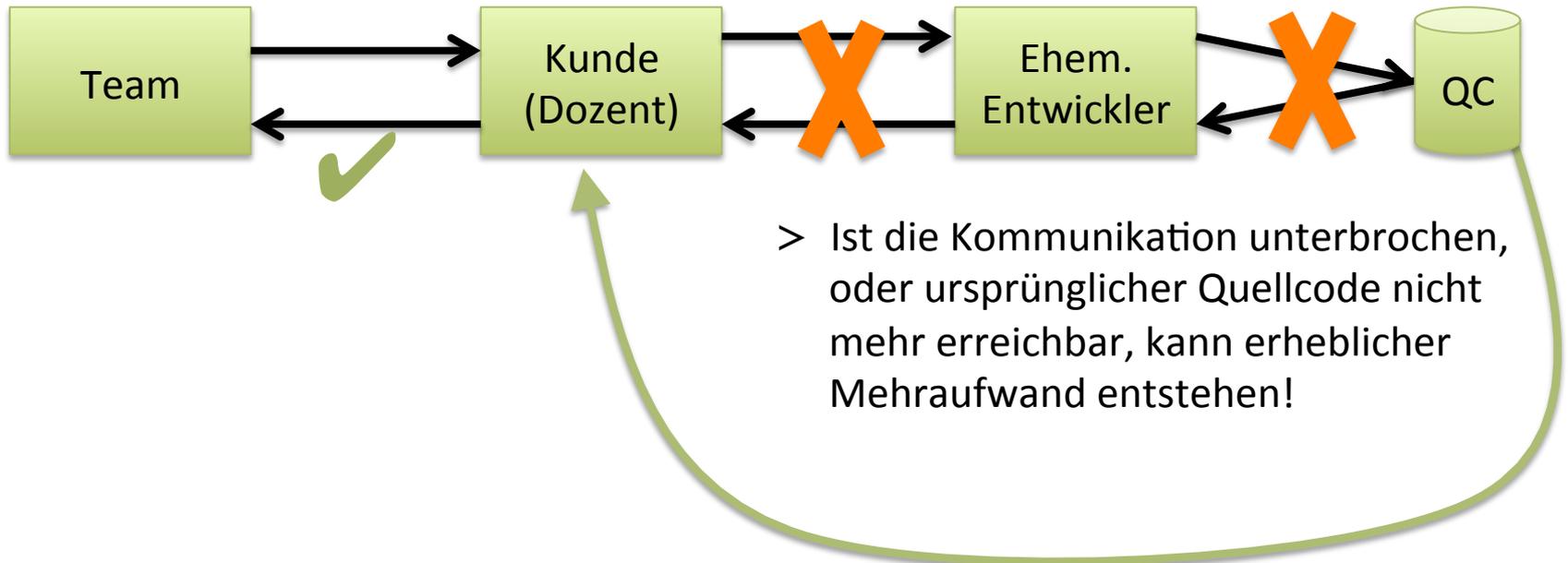


Problem: Code-Differenzen

... so sah es aus!

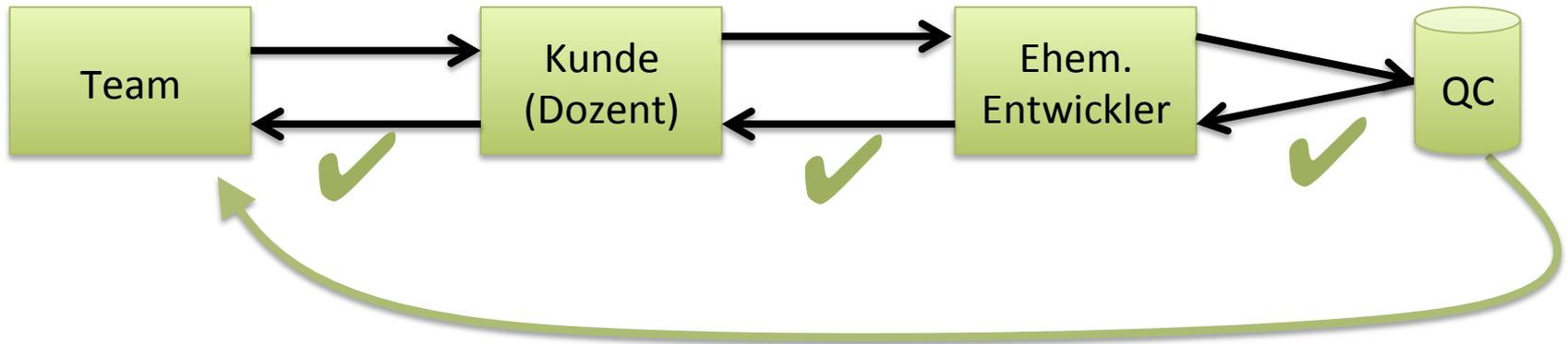


Problem: Code-Differenzen



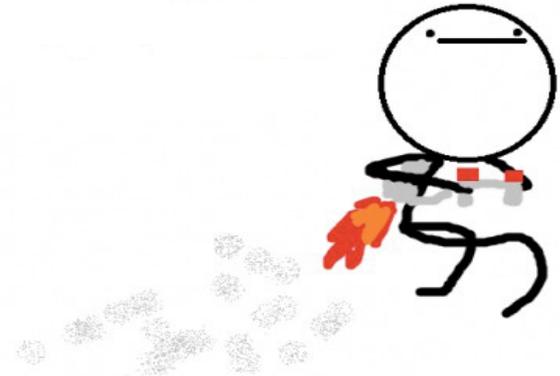
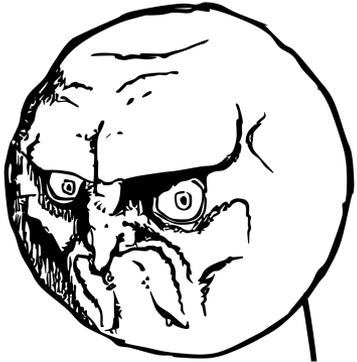
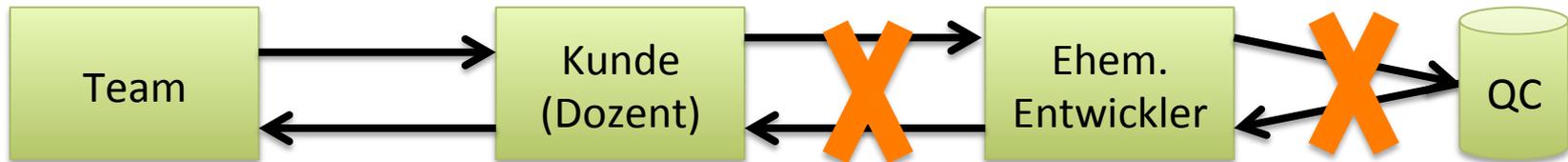
- > Ist die Kommunikation unterbrochen, oder ursprünglicher Quellcode nicht mehr erreichbar, kann erheblicher Mehraufwand entstehen!
- > Die endgültige Quellcode muss dem Kunden am Ende des Projektes vorliegen.
- > Dieser ist auch dafür zuständig, ob das Projekt aus dem Code generiert werden kann (wenn nicht Vertraglich festgehalten).

Lösung: Kontakt mit ehemaligen Entwicklern



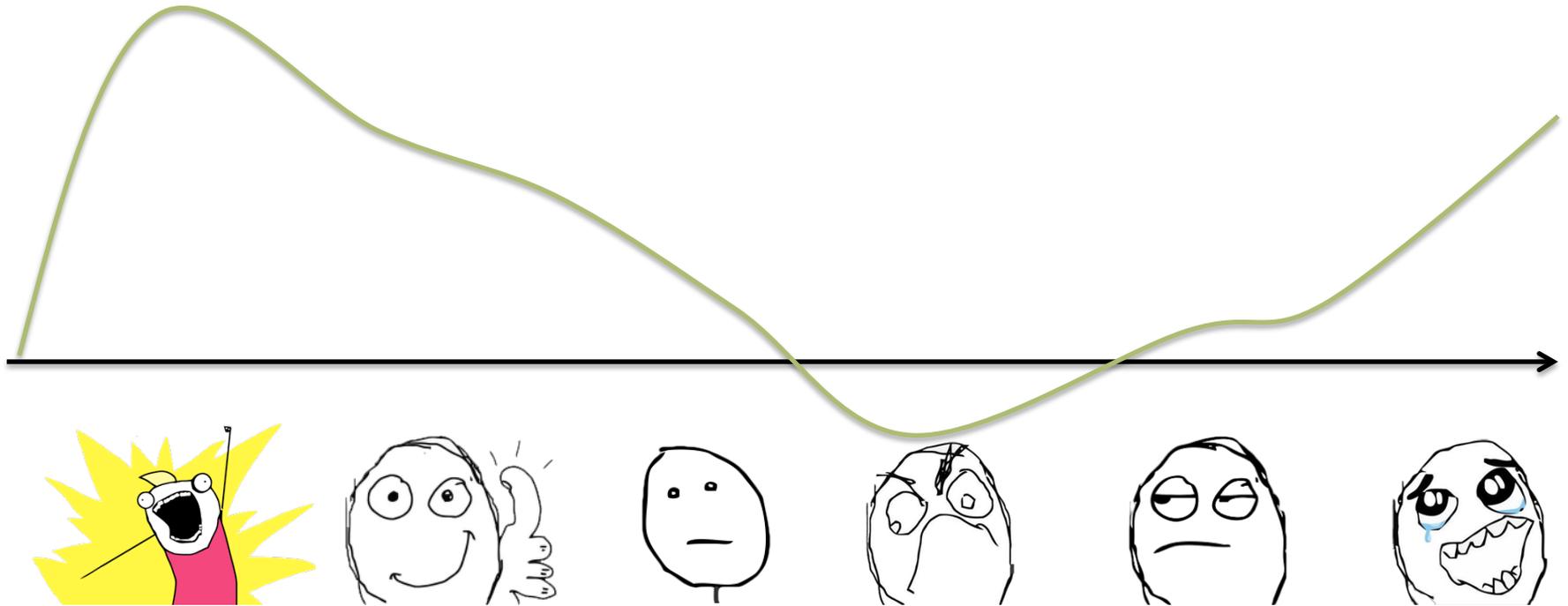
Im Optimalfall kann der Kontakt mit der ehemaligen Entwicklern hergestellt werden und das Problem besprochen werden.

Lesson Learned: Fehlerhafter Quellcode kann das Projekt stark behindern

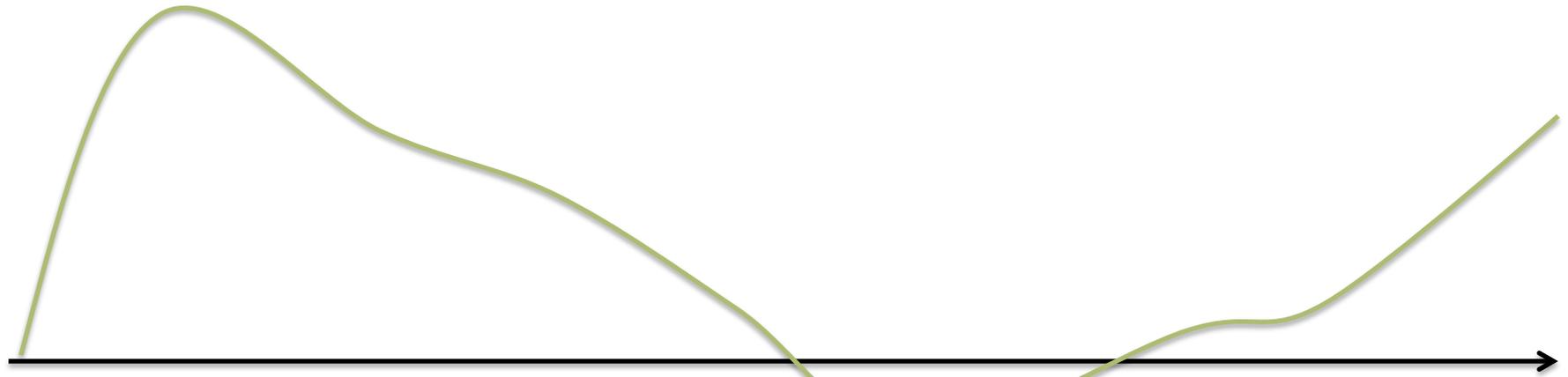


Ansonsten ist das Team aufgeschmissen,
der Kunde weiß leider auch nicht weiter
und das ehemalige Entwicklerteam hat sich bereits aus dem Staub gemacht!

Problem: Stagnierende Motivation gegen Mitte-Ende des Projektes



Lesson Learned: Nutzen der Motivationskurve



- > Anfang nutzen um in den Möglichkeiten zu schwelgen
- > Größte Kreativität am Anfang
- > Ausführliches Brainstorming - Erfassen aller Gedankenschritte

- > Aktiv die Zuständigkeiten aufspalten und grundlegende Arbeitspakete verteilen

- > Entspanntes Teammeeting um das Team zu pushen und neue Motivation zu geben

- > Projektkoordination intensivieren, Fehler und Bugs suchen, Arbeitspakete mit knappen Deadlines setzen

Agenda

- > Schwarmintelligenz (25')
- > Ausgangspunkt (5')
- > Softwareentwicklung (30')
 - Problemstellung
 - Herangehensweise
 - Probleme, Lösungen, Lessons Learned
 - **Beispiel**

Resümee Was hat er nochmal gesagt?

Ameisenalgorithmus

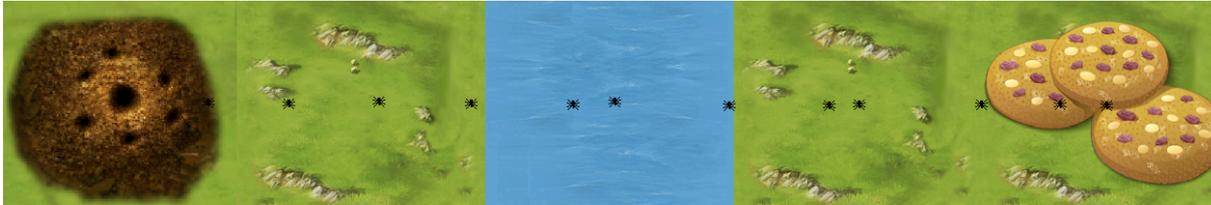
- > Einzelne unintelligente Individuen laufen auf einem Graphen
- > Kommunikation erfolgt durch Umgebungsveränderung (Pheromone)
- > Stochastische, kurzsichtige Entscheidungen auf Basis Pheromonkonzentration
- > Anzuwenden bei dynamischen Problemen
- > Bei statischen Problemen sind statische Algorithmen zu bevorzugen (Dijkstra)
- > Anwendungen in Bereichen wie Navigation, Logistik, Spielen und Optimierungen

Softwareentwicklung

- > Gute Strukturen gewinnen; Dokumentation ist unerlässlich
- > Abstraktion mit Sinn und Verstand
- > Verteilung der Aufgabenbereiche nach Klassen
- > Offenes Ohr und Augen für Probleme
- > Eine endgültige Endversion abliefern
- > Motivationskurve beachten und nutzen

Start!

Abluss Weiterentwicklungsmöglichkeiten



- > Schnellere Reaktion auf Blockaden
- > Wirkliches Pausieren der Simulation
- > Optimierung des Verhaltens bei mehreren Zielen (Keks)
- > Statistik zur Auswertung beim Beenden des Algorithmus
- > Fehler beim Laden von Karten beheben

Danke für die Aufmerksamkeit!

Lennart Steffin

WInf1026@fh-wedel.de

<http://stud.fh-wedel.de/~winf1026> (wird bald erstellt)

Download des Simulators unter:

<http://www.fh-wedel.de/mitarbeiter/iw/software/>