

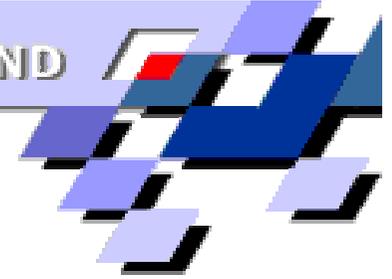
CI in games -

Methoden der Computational Intelligence zur Entwicklung von Spielstrategien

Nicola Beume Boris Naujoks Mike Preuß



Lehrstuhl für Algorithm Engineering (Ls11)
Fachbereich Informatik
Universität Dortmund



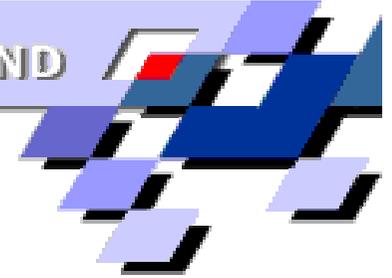
Infoveranstaltung

➤ Ablauf

1. Kurzvorstellung der Veranstalter
2. Präsentation (Inhalt, Ablauf, etc. der PG)
3. Demo
4. Ehemalige PG-Teilnehmer berichten
5. Fragen

➤ Weitere Informationen

<http://ls11-www.cs.uni-dortmund.de/people/beume/pggames/>
Übersichtsartikel zum Mitnehmen



Computational Intelligence

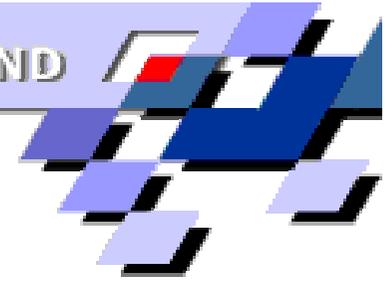
- Als Gegenansatz zur 'klassischen' symbolischen Artificial Intelligence (künstlichen Intelligenz)

Natur-inspirierte Verfahren zum Optimieren, Steuern, Lernen

- Evolutionäre Algorithmen
- Fuzzy Systeme
- Schwarm Intelligenz (Ameisen-Kolonien, PSO)
- Künstliche Neuronale Netze

EA, KNN, PSO randomisiert

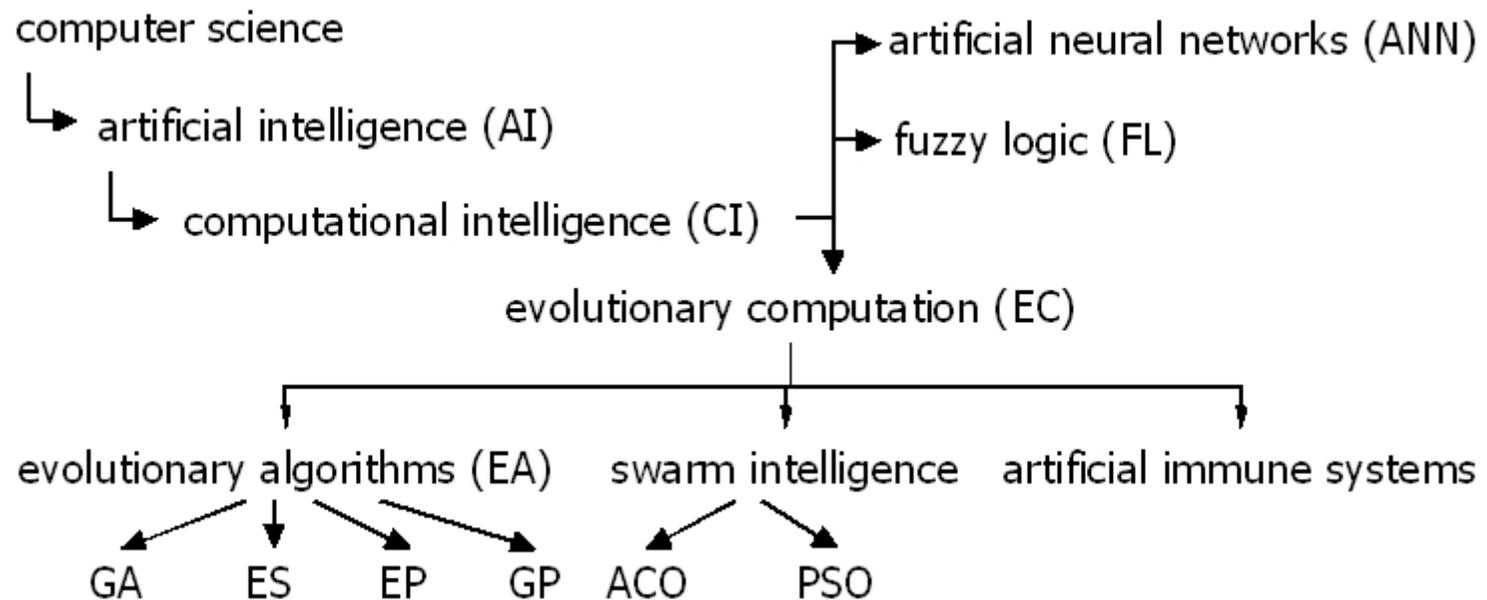
Fuzzy Systeme verarbeiten unscharfe Informationen (z.B. Wahrscheinlichkeiten)

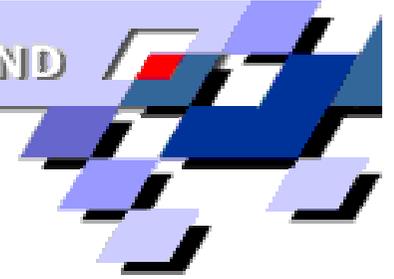


Computational Intelligence

“These technologies of neural, fuzzy and evolutionary systems were brought together under the rubric of Computational Intelligence, a relatively new field offered to generally describe methods of computation that can be used to adapt solutions to new problems and do not rely on explicit human knowledge.”

David Fogel





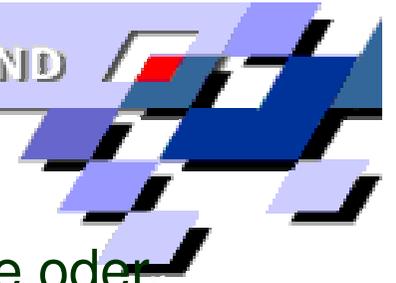
CI in games

- Ziele:
 - Computerspieler lebendig machen (lernfähig, unberechenbar)
 - mehr Spielspaß

- 5 zentrale Probleme:
 - Bewegung: Wohin gehe ich überhaupt?
 - Wege finden: Wie komme ich da hin?
 - Taktische Analyse: Wie ist die Situation, welche Möglichkeiten habe ich?
 - Entscheidungsfindung: Welche Alternative wähle ich?
 - Lernen: Wie muss ich mein Verhalten anpassen?

Anwendungsgebiete:

- Testen von Spielen (während Entwicklung)
- Intelligenz für NPCs

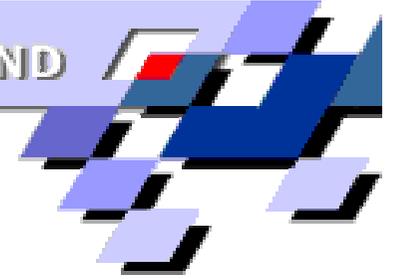


Warum KI?

Bisher zumeist statische oder fuzzybasierte Regelsysteme oder Scripting als 'Kern' der Spiele-KI

- wenig Veränderung, trotz dramatischer Hardware-Entwicklung





Kategorien von Spielen

- Spiele mit vollständiger Information (Schach)
 - lösbar mit viel Rechenkapazität, Datenbanken
- **Spiele mit unvollständiger Information** (Poker)
 - Spielbaum zu groß für vollständige Berechnung
 - Wahrscheinlichkeiten abschätzen
- “Real-World”-Spiele (Fußball)
 - zu komplexe technische Problem für PG
- **Video-Spiele** (Pac-Man, Siedler,...)
 - aktuell typisch: Verhalten von NPCs deterministisch
 - Intelligenz durch “Mogeln” (zusätzliche Informationen, Ressourcen)
 - Einsatz von Heuristiken vielversprechend

Pac-Man

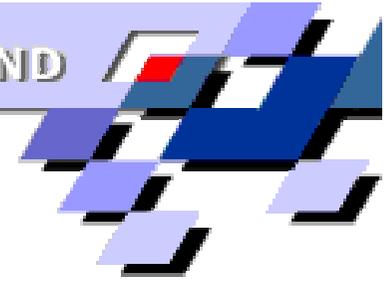
z.B. Yannakakis (2005): Generating Interesting Interactive Pac-Man Opponents

- Wie misst man interessant (automatisierbar)?
- Aus Sicht der Gegner: Interessant ist nicht gleich optimal!
- Eine optimale Gegnerstrategie erledigt Pac-Man schnell und effektiv...

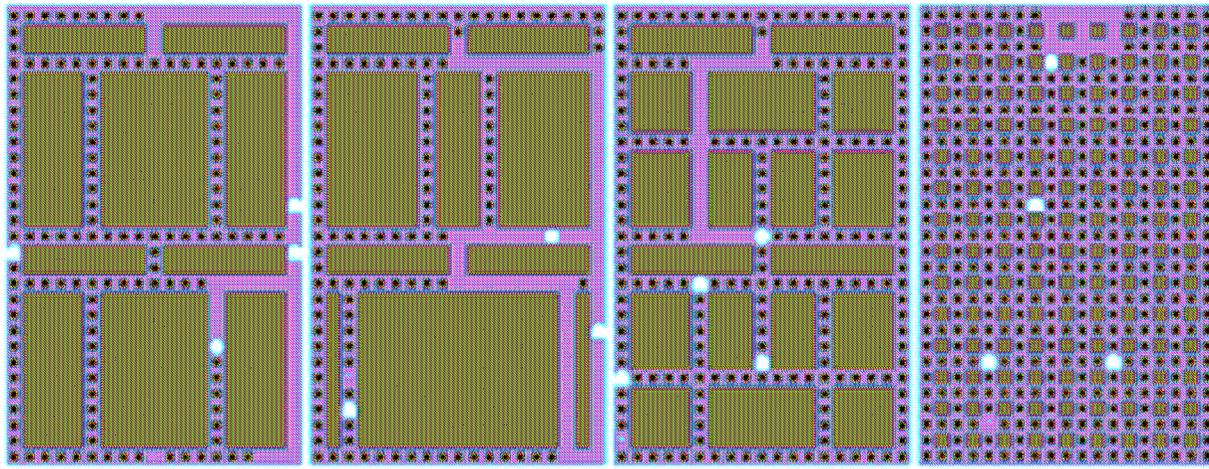
Mögliche CI-Ansätze:

- Adaptives Regelsystem
- (Adaptives) künstliches neuronales Netz
- Genetic-Programming, Learning classifier...
- Fuzzy-Controller, ... Hybrid...





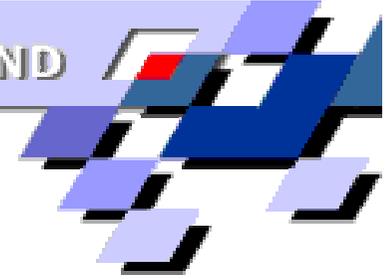
Pac-Man Strategien



Aus der Sicht der Geister:

- Spielfelder mit großen Ganglängen sind 'leichter'
- 3 Standardstrategien:
 - Zufallsbewegung
 - Abstand zum Pac-Man verringern
 - Kraftmodell: Ausserdem Abstand zwischen Geistern erhöhen

Nero



Neuro-Evolving Robotic Operatives (NERO) is a unique computer game that lets you play with adapting intelligent agents hands-on. Evolve your own robot army by tuning their artificial brains for challenging tasks, then pit them against your friends' teams in online competitions!

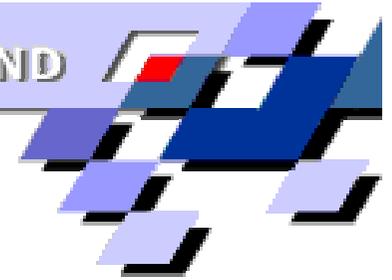
Zwei Phasen:

1. Training
 - Spieler trainiert Gruppen von Robotern
 - Teams für Duelle aus mehreren Gruppen
2. Duell/Wettkampf

Teams treten gegen andere Teams an:

- Computer-gesteuert
- LAN
- Internet





Nero

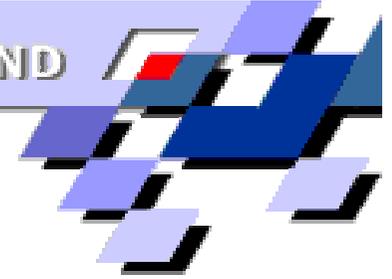
Lernalgorithmus: **rtNEAT**
(real-time *Neuro-Evolution of Augmenting Topologies*)

Mögliche Ansatzpunkte für PG:

- Alternative zu rtNEAT
- Intelligente Strategien für Gruppen
- Intelligente Strategien für Gegner
- Mehrere Szenarien?!?

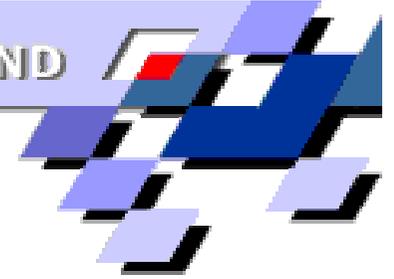
Demo-Video





Ablauf der PG

- Anfang SoSe07: Seminarphase (Universitätskolleg Bommerholz)
- einfaches Spiel: [Pac-Man](#)
- Spielstrategien für mehrere Spielfiguren: [Spieler, Geister](#)
- Spielstrategien mit mehreren CI-Methoden
- Vergleich nach mehreren Kriterien: [Spielspaß, Spielerfolg](#)
- Übertragung der Methoden auf komplexeres Spiel: [NERO](#)
- Evaluation in Spielturnieren, Messung und Auswertung sind allerdings nichttrivial (Messung von Spielspaß)



Anforderungen an Teilnehmer

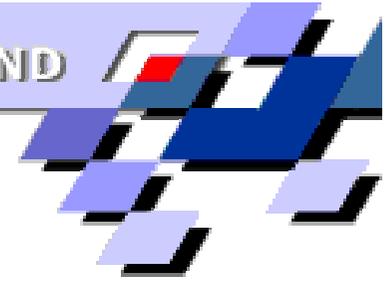
Motivation, Kreativität:

- Kreativer 'Eigenanteil' der PG-Teilnehmer notwendig, da nicht abzusehen ist, was gut funktionieren wird (Machbarkeitsstudie)
- Selbstständiges Arbeiten zwingend erforderlich

Entwickeln/Entscheiden im TEAM:

- Betreuer haben Interesse am Thema, sind aber nicht 'Bestimmer'
- Die meisten Großprojekte in der Informatik scheitern nicht an fachlichen Mängeln, sondern an 'Social Skills'

- CI-Methoden
- Objekt-orientierte Programmierung



Kontakt

Lehrstuhl 11, OH14 (2. Stock Neubau)

- Nicola.Beume@uni-dortmund.de
- Boris.Naujoks@uni-dortmund.de
- Mike.Preuss@uni-dortmund.de

Raum 233, Tel.: 755-7704

Raum 234, Tel.: 755-7705