

Spel-AI

**AI, fysik och animation
Tillståndsmaskiner
Världsrepresentation
Metriker
Influence maps
Flocking**

Ingemar
Ragnemalm
ingis@isy.liu.se

Labbarna

E-posthjälp såväl som i labbet

**Labb 3 har komplext labbskal - fråga
hellre än svettas!**

**Labb 4 kommer - och skalet blir riktigt
litet.**

Projekten

**Preliminärt förslag senast torsdag.
Behöver inte se ut som en spec.**

Spec inne till lördag.

**Gör något intressant, men inte något
alltför komplext!**

Projektspecen

Kortfattad!

Titel + Vem

Sammanfatta vad som skall göras

Lista av skall- och börförkrav.

Vari består milstolpen?

Spel-AI och spel-fysik

Tillämpningar av fältens grundläggande tekniker.

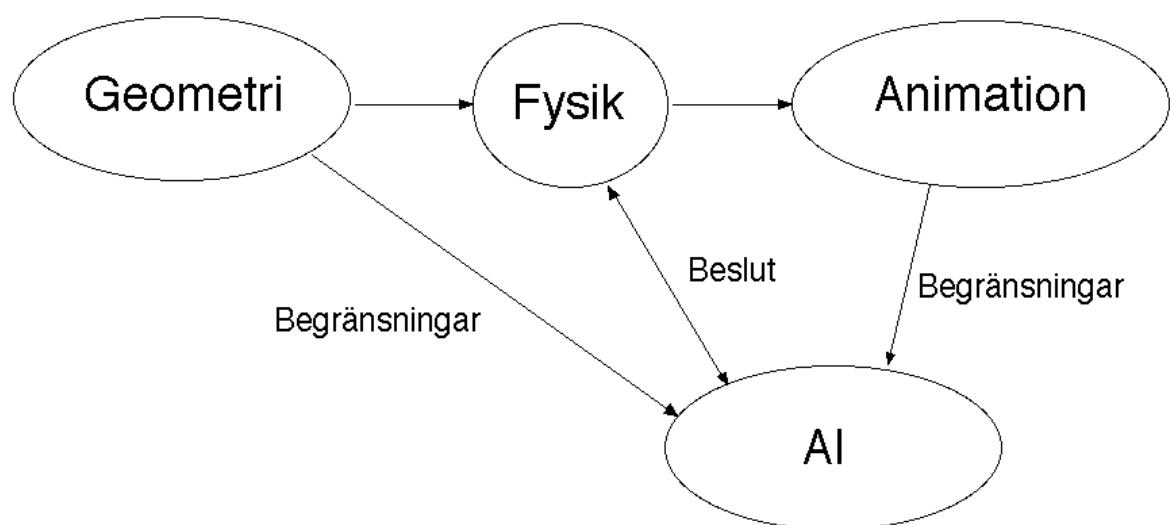
AI: Planering, beteenden, beslutsfattande

Fysik: Snarast mekanik och en del klassisk fysik

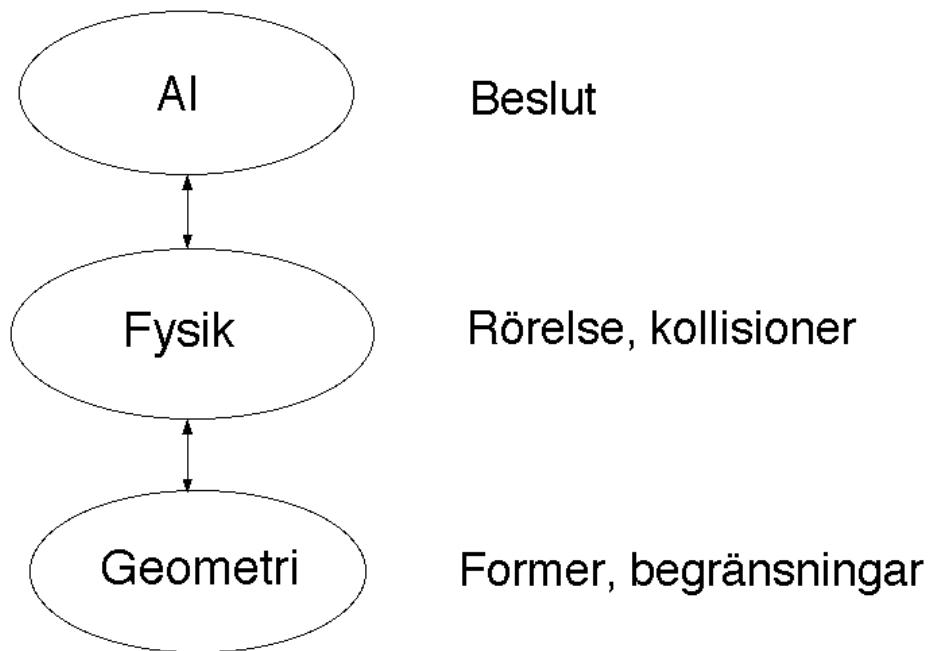
Animationer är i hög grad resultat av dessa.

Ingemar
Ragnemalm
ingis@isy.liu.se

AI, fysik och animation i spel



Ett system i flera lager



Spel-AI

Äntligen en “artificiell intelligens” som vanligt folk begriper! Ett genombrott för hela AI-begreppet?

Typiska spel-AI-problem

- Planering (vägsökning)
 - Beteenden
 - Beslutsfattande

Ingemar
Ragnemalm
ingis@isy.liu.se

Två sorters spel-AI

- Agenter: fiender och andra datorstyrda spelobjekt
- Abstrakta kontrollsysten, övergripande beteenden hos t.ex. en armé eller en skock får

Ingemar
Ragnemalm
ingis@isy.liu.se

Viktiga problem att ta hänsyn till:

Spatiell information

Världens geometri, andra agenter, spelaren, tittrikningen

Minne

Hur lagrar vi det agenten skall komma ihåg?

Analys

Ofta släende enkla metoder!

Aktion

Vad skall agenten utföra? Förflyttningar mm.

Fusk

Bör agenten tillåtas fuska?

Vanliga metoder

- Finita tillståndsmaskiner
 - Regelsystem
- Planering, kortaste-vägen-problemet
 - Minmax-analys

Typisk actionspels-AI

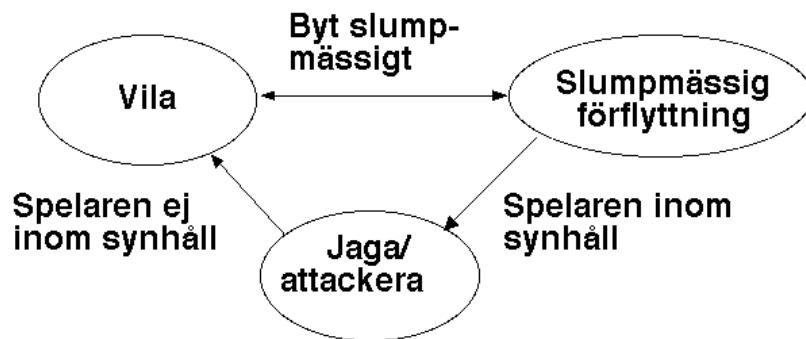
Vanliga beteenden:

- Jaga
- Fly
- Slumpmässig förflyttning
- Följ en bana
- Vila

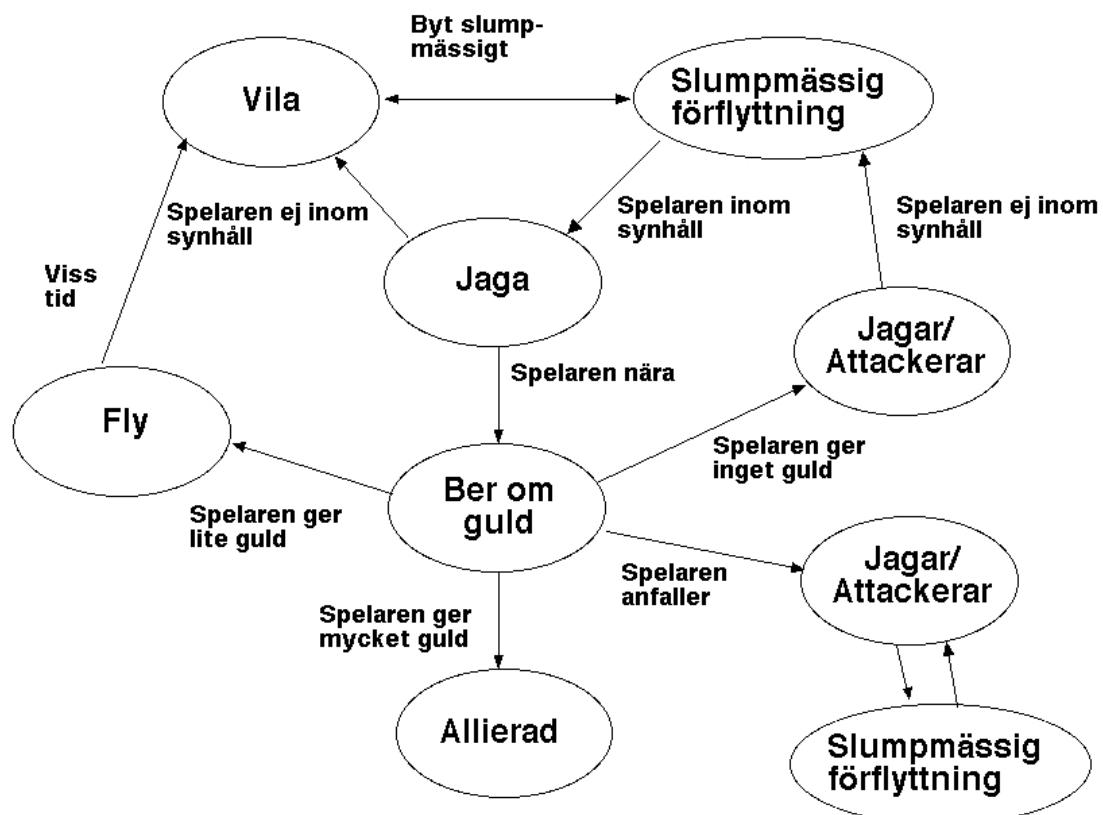
Agenten byter mellan dessa beteenden
beroende på spelhändelser:

- Ljud i närheten
- Spelaren/intressant objekt inom synhåll
- Agenten skadas, angrips
- Kommunikation

Detta kan göras med en finita
tillståndsmaskin (FSM). Enkelt exempel:



Något mer avancerat exempel



Parallelta FSM

Finita tillståndsmaskiner är ofta onödigt formella och onödigt stelbenta

Parallelta FSM ger mer frihet och är ofta enklare än FSM. Flera delbeteenden kan modelleras separat.

Regelsystem

**Enkla regelsystem kan representera
många action-AI**

- Om jag inte ser någon spelare, vandra slumpmässigt
- Om jag ser en spelare, gå ditåt
- Om jag är nära spelaren, anfall

Fler action-AI-problem

Ögonkontakt, synfält

Hide&take cover

Jaga med prediktion

Skjuta (ev. med prediktion)

Världrepresentation

Viktigt problem för AI! Det viktigaste?

**Många AI-problem lösas genom att välja
världrepresentation**

- Hur ser AI'n geometrin?
 - Hur lösas synlighet?
 - Ger vi för mycket information till agenterna, så de kan fuska?
 - Annan extra information?

Ingemar
Ragnemalm
ingis@isv.liu.se

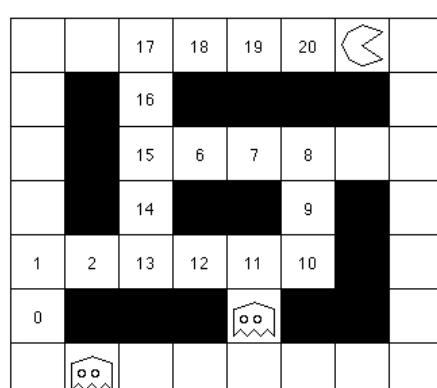
Världrepresentation

Exempel:

Pac-Man-likt spel

Världen är en grid, representeras med array

Extra information: Spelaren lämnar "doftspår" efter sig som fienderna kan följa



Världrepresentation

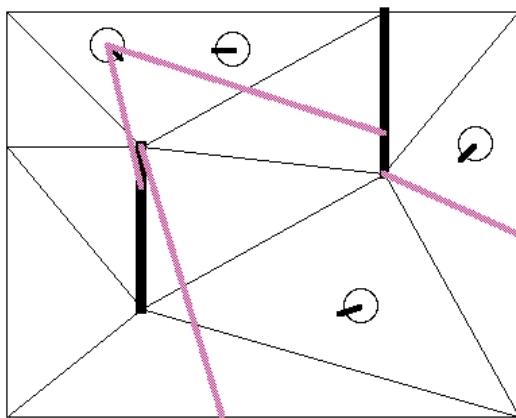
Exempel:

Robowar-likt spel men med hinder

Världen är en polygonmesh av trianglar

Trianglarna har pekare till alla grannar och pekare till alla objekt i triangeln

Synlighet i övriga trianglar kan bestämmas med portal-likt
frustumberäkningar (men i 2D)



Avstånd

Världrepresentationen är ofta gridbaserad!

Diskret värld -> avstånd ofta icke-Euklidiska!

Exempel:

Schack

Sid Meier's Civilization

Metrik

En metrik är ett avståndsmått som uppfyller:

$$\begin{aligned}D(p_1, p_1) &= 0 \\D(p_1, p_2) &> 0, p_1 \neq p_2 \\D(p_1, p_2) &= D(p_2, p_1) \\D(p_1, p_3) &\leq D(p_1, p_2) + D(p_2, p_3)\end{aligned}$$

Metriker

Euklidisk: $D = \sqrt{(\Delta x^2 + \Delta y^2)}$

City Block: $| \Delta x | + | \Delta y |$

Chessboard: $\max(| \Delta x |, | \Delta y |)$

Viktade heltalsmetriker:

**Chamfer 2-3
Chamfer 3-4
Chamfer 5-7-11**

Viktade flyttalsmetriker

Chamfer 1- $\sqrt{2}$

Taktisk AI

Planering av förflyttning

“Crash & turn”

Dijkstras algoritm

A*

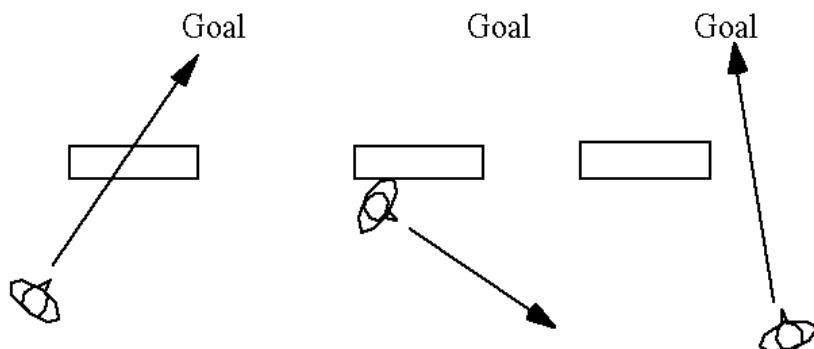
Omplanering när en väg blir oframkomlig

Är den optimala vägen rätt?

Ingemar
Ragnemalm
ingis@isy.liu.se

“Crash & turn”

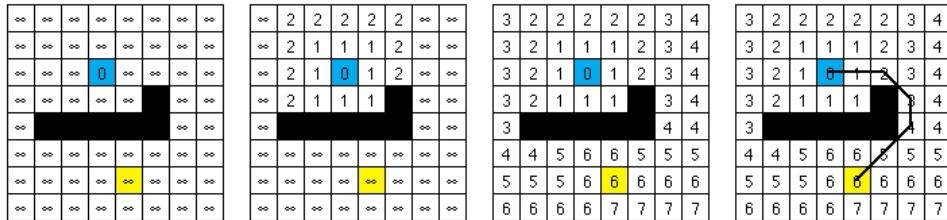
- Gå mot målet tills nånting tar emot
 - Gå i slumpmässig riktning
 - Upprepa



Fångit? Planlöst? Det beror på vilket beteende som agenten bör ha!

Kortaste-vägen-problemet

En geometri representerad som regelbunden grid: Löses med avståndstransform



(OBS! Vilken metrik används?)

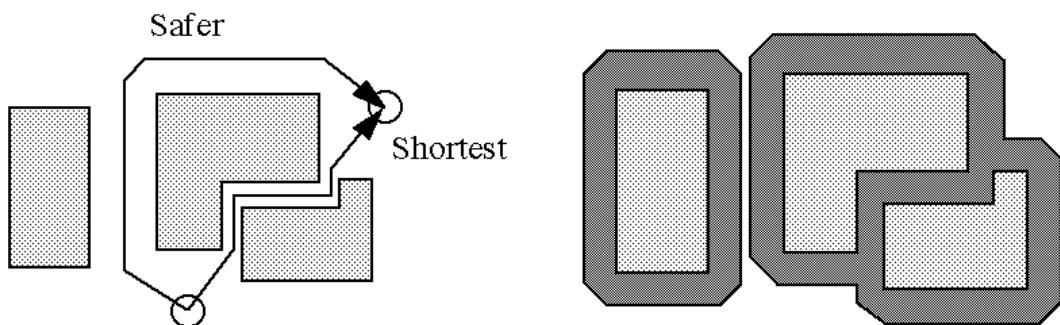
A* lägger till heuristiska mått för att söka åt troligaste hålet först

Men är detta egentligen AI? Och är det den kortaste vägen vi vill ha?

Säkraste vägen?

Prioritera vägar beroende på

- hur trångt det är
- faror, fiender...



Breda vägar kan hittas med matematisk morfologi - expansion av hindren

Sökning av tillståndsrymden

Minmax, Negamax

Metoder speciellt lämpade för deterministiska brädspel

Sök den där mitt bästa drag är bäst och motståndarens bästa drag är sämst

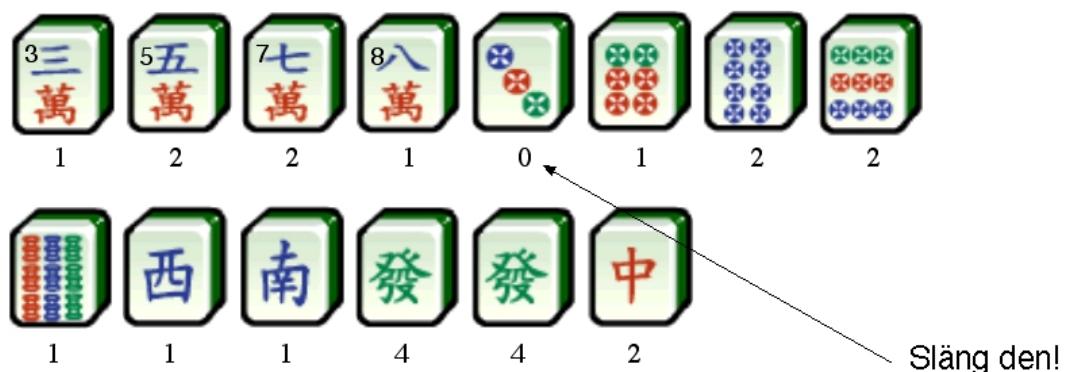
alpha-beta-pruning begränsar sökrymden till de delar som bedöms viktigast

Ingemar
Ragnemalm
ingis@isy.liu.se

Heuristisk utvärderingsfunktion

Lämpat för bl.a. icke-deterministiska brädspel

Gammalt favoritexempel:



- +1 för honör, drake, vind
- +2 om drake eller egen vind
- +2 för varje likadan (möjlig pong)
- +1 för granne inom två steg (möjlig chow)

Influence maps

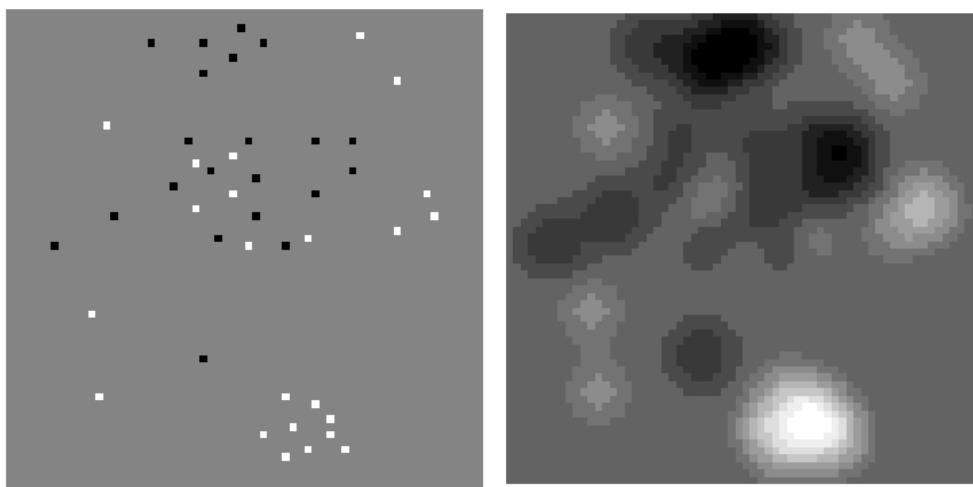
Kraftfull representation av information om spelsituation!

Lågpassfilter av karta av spelagenter förenklar analysen.

- Vem “äger” rutan?
- Är området säkert, inga fiender nära?
- Vilken annan medspelare står fri?
- Var är motståndaren som svagast?

Ingemar
Ragnemalm
Inglis@isy.liu.se

Influence maps, exempel



Fatta beslut beroende på influensvärdet

- Anfall svaga punkter
- Passa till obevakade medspelare

Varianter

- Terrängberoende filtrering, längsammare spridning i svår terräng
 - Flera influence maps för olika data
 - Olika värde i kartan för olika pjästyper

Varianterna styrs i stort av speltypen, men grundprincipen är generellt användbar till många problem.

Influence maps och bildanalys

En influence map är en *bild* av den spatiella situationen.

Bildanalys är ett välutforskat ämne - många metoder finns för att analysera innehållet.

- Segmentering, watersheds
- Linjära och olinjära filter
- Formanalys, skelettering

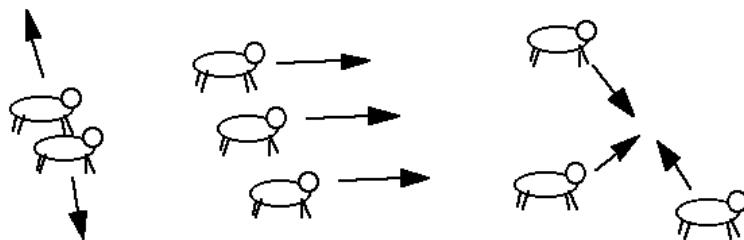
Analysen kan lätt snabbas upp på GPU med vanliga grafikmetoder

Flocking - Boids

Gruppbeende kan simuleras med ganska enkla medel.

Craig W Reynolds "boids", baserade på tre enkla regler:

- Separation - undvik att gå för nära varandra
 - Alignment - sträva åt samma håll
 - Cohesion - sträva mot mitten av gruppen



Boids, implementation

Partikelsystem

Separation: Kraft bort från närmaste boid

Alignment: Beräkna medelhastighet, drag egen hastighet mot denna

Cohesion: Beräkna medelposition, kraft mot denna

Boids

```
for all boids i
    for all boids j
        if dist(i, j) < kMaxDistance
            count += 1
            accumulate speed difference
            accumulate average position
            accumulate avoidance vector

    if count > 0 then
        divide accumulated data by count

for all boids i
    affect speed by
        speed diff (alignment)
        average pos (cohesion)
        avoidance vectors (avoidance)
    position += speed
```

Viktiga parametrar att "skruva på"

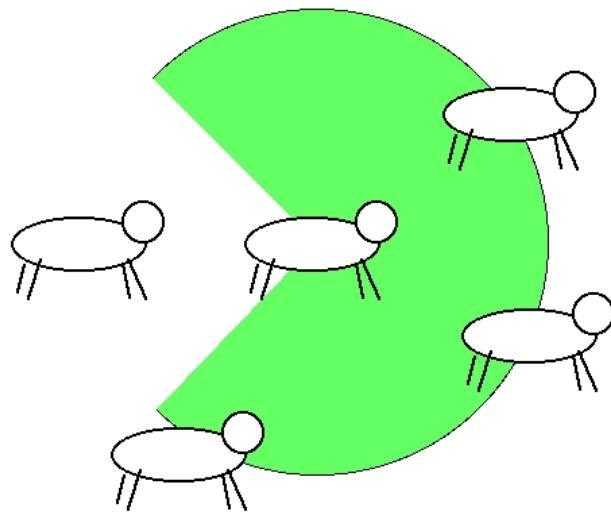
Vilken vikt på varje komponent?

Hur stor omgivning skall räknas?

Avståndsberoende funktioner för separation och cohesion?

Boids, synfält

En boid kanske inte bör ha "ögon i nacken"

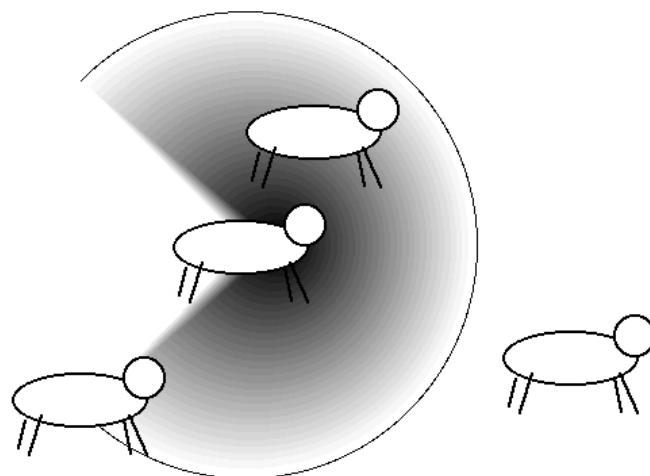


Vilka andra boids skall räknas?

Begränsa på vinkel såväl som avstånd.

Boids, funktioner

En enkel tröskelfunktion kan ge ryckig animation



Mjukt avtagande funktion?

För stark separation kan ge våldsamma krafter om funktionen blir för stor nära

Boids är exempel på A-life

Imitation av verkliga beteenden

Optimala lösningar är inte målet!

**En A-life-vägsökning får gärna vela och tramsa,
om det stämmer med agentens typ.**

**Optimal lösning för ett djur på vägen vs verkligt
beteende?**

Flockinglabben på onsdag

Baseras på enkel 2D-spritemotor

Implementera boids-algoritmen

Vissa variationer

Labbmaterialet är på gång

Lärande system

System som lär sig är ett intressant ämne! Mest kända metoderna:

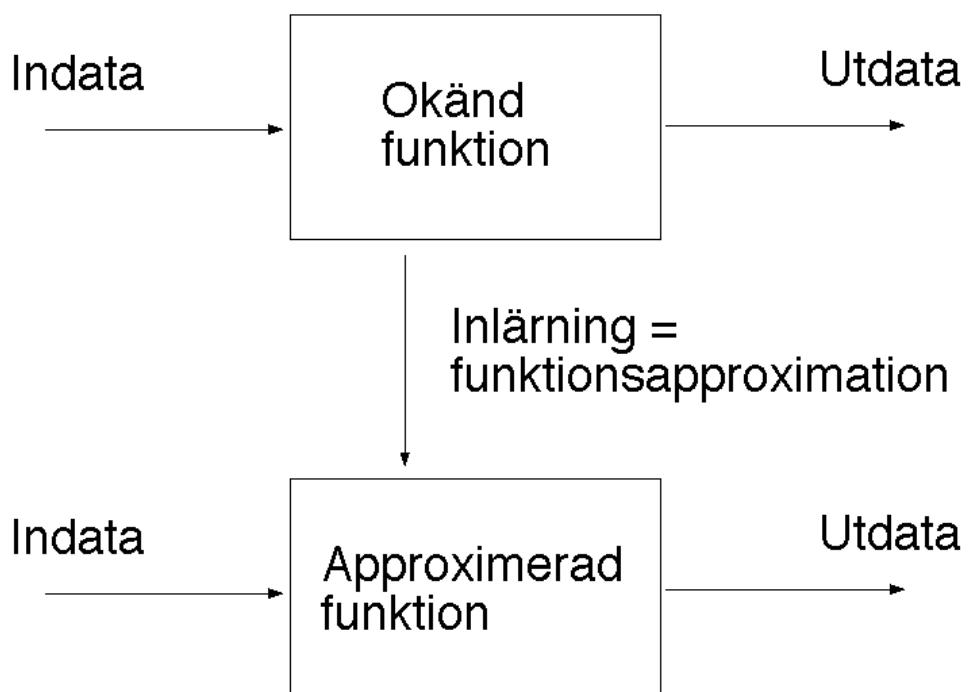
- Nervnät
- Genetiska algoritmer

Aningent mindre kända:

- Simulated annealing
 - Hill climbing
- Heuristiska metoder

Men observera: Allt är i grund och botten *optimering!*

Optimering?



Grundmetoder för lärande system/funktionsoptimering

Hill climbing

Simulated annealing

Genetic algorithms

Lärande system/ funktionsoptimering, exempel

**Minigolf: Slå bollen, få den att träffa så
nära hålet som möjligt**

Indata: Vinkel, utgångshastighet

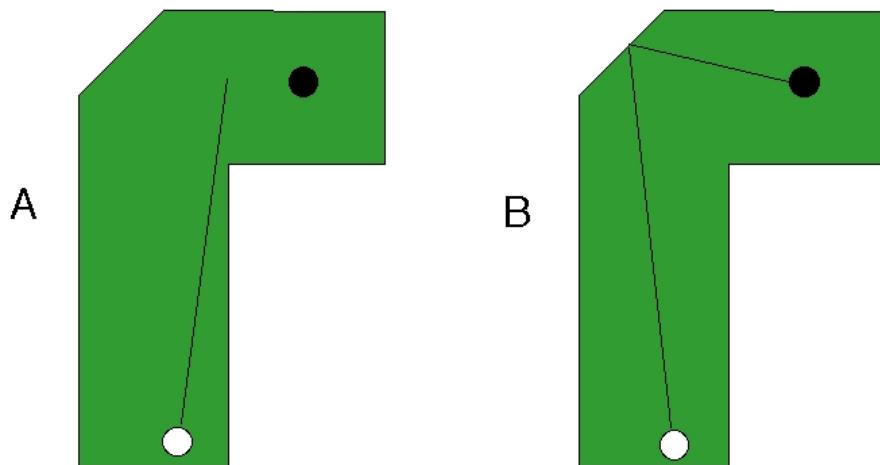
Utdata: Bollens slutposition

**Målfunktion: Avstånd mellan utfall och
mål.**

Problem: Hitta rätt indataparametrar.

Lokala minima måste undvikas

Exempel: minigolf



Små förändringar kring A ger sämre lägen; större
avstånd till hålet

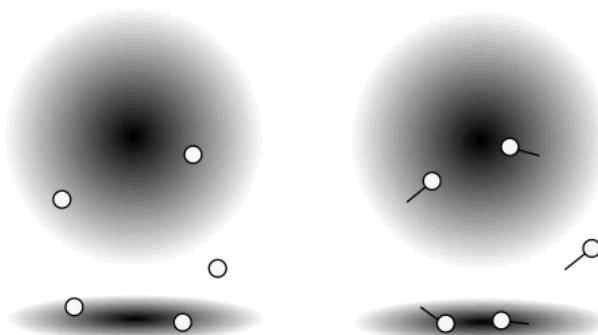
Lärande system/ funktionsoptimering, exempel

Hill climbing: Ändra parametrarna åt det håll
som borde bli bättre.

Simulated Annealing: Ändra parametrarna
slumpvis, behåll de bästa. Minska
slumpstegen efter hand.

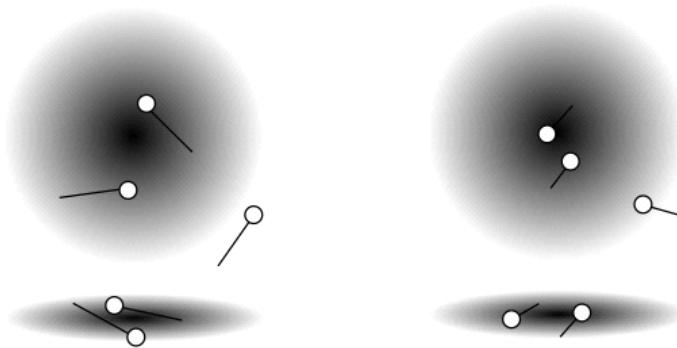
Genetic algorithms: Kombinera par av försök
till nya, behåll de bästa.

Hill climbing



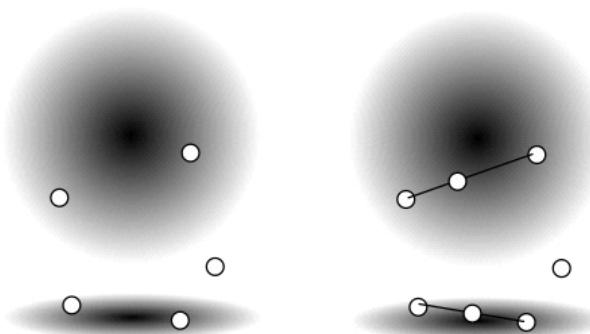
Sök närliggande bättre platser tills vi hittar lokalt minimum

Simulated annealing



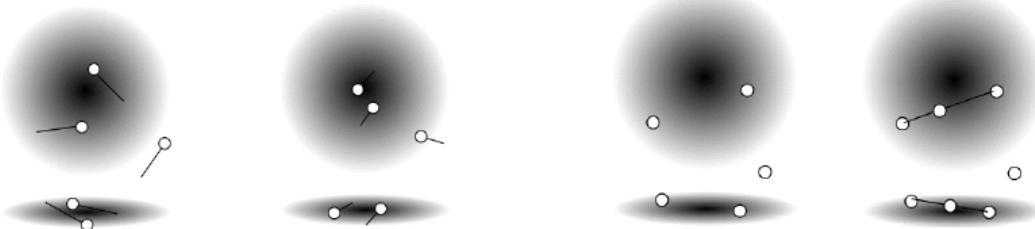
Tag mindre och mindre slumpmässiga steg

Genetiska algoritmer



Kombinerar tidigare lägen

Simulated annealing eller genetiska algoritmer?



Simulated annealing: Mer frihet att ta steg utanför tidigare sampel. Fastnar inte i felaktiga antaganden.

Genetic algorithms: Kan söka bättre längs "smala" områden, samlar sig bättre kring optimum i sådana fall.

Nervnät (Neural networks)

En representation av systemet, av den önskade funktionen.

Kan med tidigare nämnda metoder modellera komplexa funktioner.

Funktionsapproximation är inte alls tråkigt, men...

Mitt dilemma:

Är detta relevant för spel?

Både jag och Millington säger *nej*. Inlärning har en ytterst liten roll i spel!

Heuristiska metoder troligen intressantast, för begränsad inlärning hos agenter.

Nervnät mm har framgångsrikt använts på bl.a. bildanalys, men knappast på spel.

Spel med inlärning eller liknande inslag:

SimLife

Viss simulering av genetik, men

- urval av population är inte (exakt) samma sak som inlärning och upplevs inte som det
- Spelet var inte särskilt underhållande och simuleringen var dålig

Black&White

Inlärning av beteende på “varelsen”

Intressant idé, mindre lyckat resultat

Men är AI viktigt för spel?

Javisst!

- Beteenden och vägplanering
 - Världsrepresentation
- Analys av speldata i denna representation
- Komplex agentrepresentation för styrning av beteenden

Däremot är lärande mer marginellt