

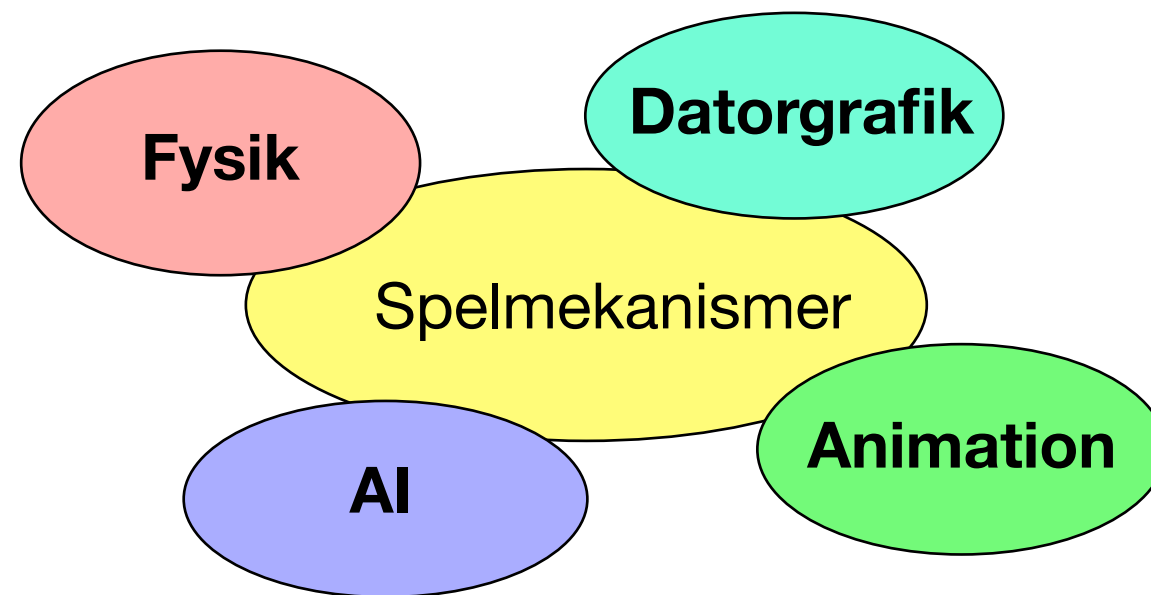


Information Coding / Computer Graphics, ISY, LiTH

# TSBK 03

## Teknik för avancerade datorspel

Ingemar Ragnemalm, ISY





Information Coding / Computer Graphics, ISY, LiTH

# Labbarna

E-posthjälp såväl som i labbet

Möjlighet att göra om **ALLA** duggor på **FÖRSTA** reservlabben  
(10:e oktober).

Man kan inte förlora poäng på att göra om en dugga. Den bästa räknas.

Omduggan görs på samma villkor som de ordinarie, upp till 5 poäng per styck. (Fast inte samma frågor.)



Information Coding / Computer Graphics, ISY, LiTH

# Projektet

Preliminärt förslag bör finnas tisdag 1/10.  
Behöver inte se ut som en spec.  
Kan vara muntligt.

Spec inne till onsdag 2/10.

Gör något intressant, men inte något alltför  
komplext!



Information Coding / Computer Graphics, ISY, LiTH

# Projektspecen

Kortfattad!

Titel + Vem

Sammanfatta vad som skall göras

Lista av skall- och börkrav.

Vari består milstolpen?



Information Coding / Computer Graphics, ISY, LiTH

## Tredjepartsverktyg: Exempel

Ljud och musik: OpenAL, SDL, FMOD

3D-grafik och 3D-spelmotorer:

Blender, Torque, Quake, Unreal, Unity, Godot... (MÅNGA!!!)

2D-spelverktyg: GameMaker, Unity, Godot, PyGame, Defold  
mm

Fysik: ODE, Newton, Bullet...



## **Tredjepartsverktyg: Varning**

Tredjepartsverktyg kostar ofta mer än de smakar!

Exempel från tidigare: Vandrande djur.

Angreppssätt 1: Fysikmotor. Svårt! Slogs mot motorn, gick inte med hjälp av den!

Angreppssätt 2: Eget enklare system anpassat för problemet. Mycket lättare!



# Dokumentera!

Vilka tredjepartsverktyg använder du?

Vilka källor stödjer du dig på?

Finns överlapp med andra kursprojekt? I så fall, vad hör till vilket projekt?



Information Coding / Computer Graphics, ISY, LiTH

# Föreläsning 10: Spel-AI

Tillståndsmaskiner  
Världsrepresentation  
Metriker  
Influence maps  
Flocking





# Spel-AI

Var en gång en “artificiell intelligens” som vanligt folk begriper! Ett genombrott för hela AI-begreppet (innan det nuvarande)?

AI var länge en besvikelse. Spel-AI och deep learning har ändrat detta.

Är spel-AI något annat än lärande system?



# Typiska spel-AI-problem

- Planering (vägsökning)
  - Beteenden
- Beslutsfattande



## **Två sorters spel-AI**

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



Information Coding / Computer Graphics, ISY, LiTH

## Viktiga problem att ta hänsyn till:

### Spatiell information

Världens geometri, andra agenter, spelaren, tittriktningen

### Minne

Hur lagrar vi det agenten skall komma ihåg?

### Analys

Ofta slående 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:

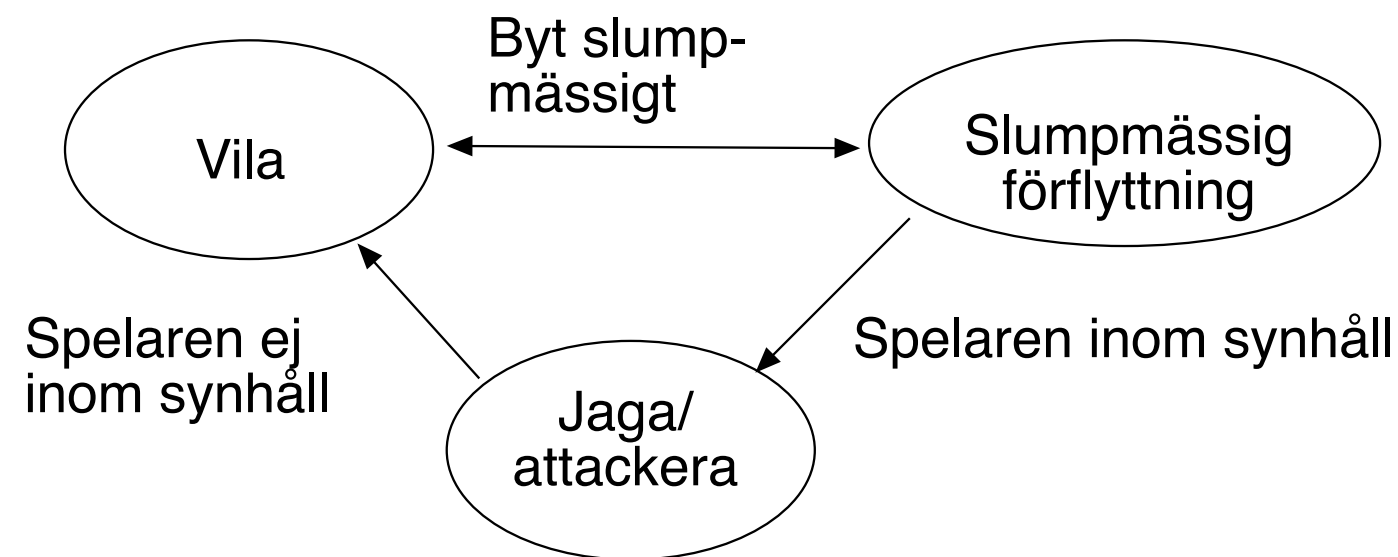
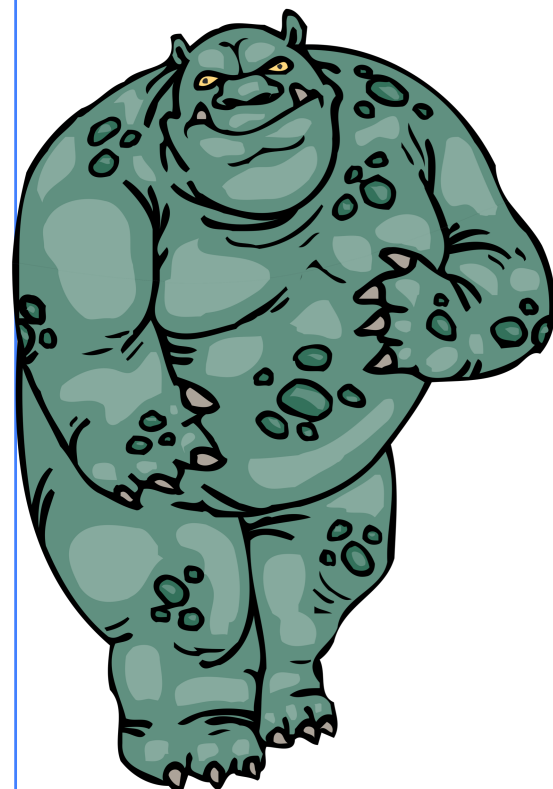
- Jaga
  - Fly
- Slumpmässig förflyttning
  - Följ en bana
  - Vila



Agenten byter mellan dessa beteenden beroende på spelhändelser:

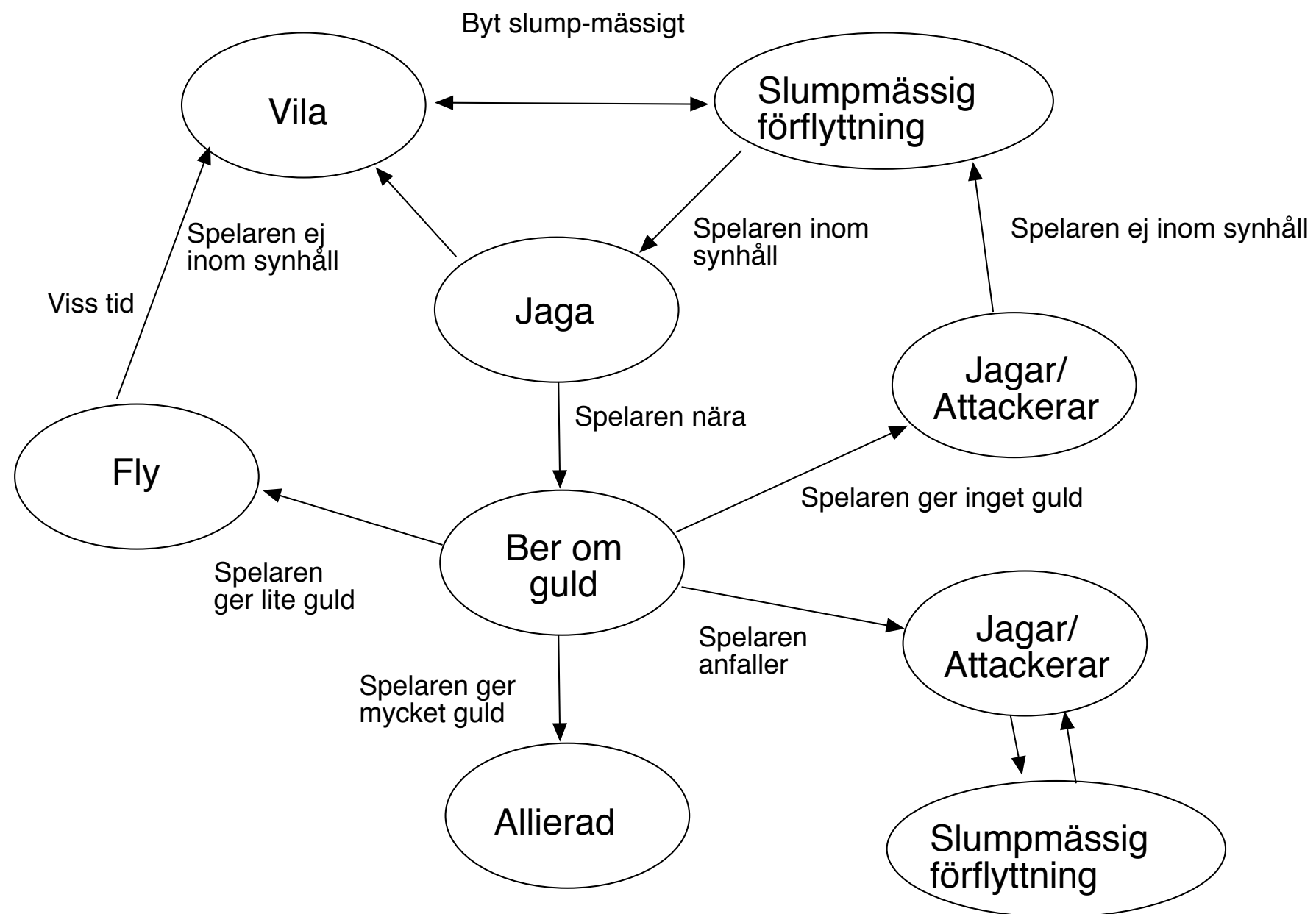
- Ljud i närheten
- Spelaren/intressant objekt inom synhåll
  - Agenten skadas, angrips
  - Kommunikation

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





## Något mer avancerat exempel







## Parallella FSM

Finita tillståndsmaskiner är ofta onödigt formella och onödigt stelbenta

Parallella 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

Minne måste lagras separat.



# 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 löses genom att välja världrepresentation

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






# Världrepresentation

Exempel:

Pac-Man-liknande spel

Världen är en grid, representeras med array

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

		17	18	19	20		
		16					
		15	6	7	8		
		14			9		
1	2	13	12	11	10		
0							
							

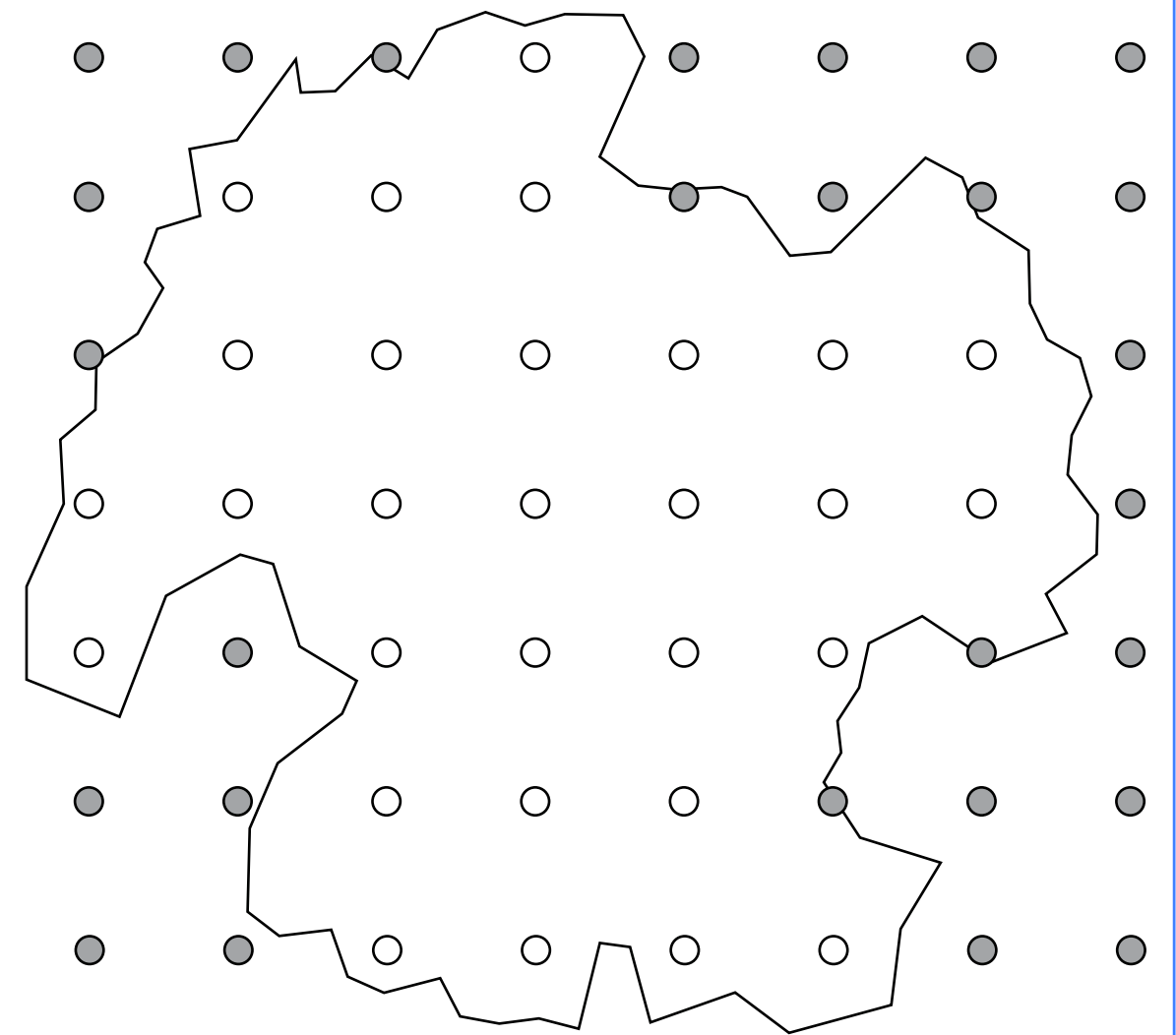


# Världrepresentation

Vad du ser och vad du "spelar" är ofta inte samma sak.

Visuellt: Mycket detaljerat, målet är realism.

AI/spelmekanismer: Lägre upplösning, förenklat, mer reguljärt.





# Avstånd

Världrepresentationen är ofta gridbaserad!

Diskret värld -> avstånd ofta icke-Euklidiska!

Exempel:

Schack

Sid Meier's Civilization (Early versions)



Definition

## 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- $\sqrt{2}$

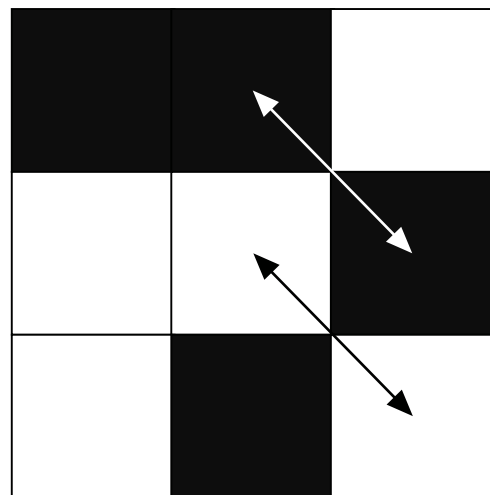


# Konnektivitet

Vad är granne till vad?

Relaterat till metriker

d4-konnektivitet för en domän ger d8 i den andra!

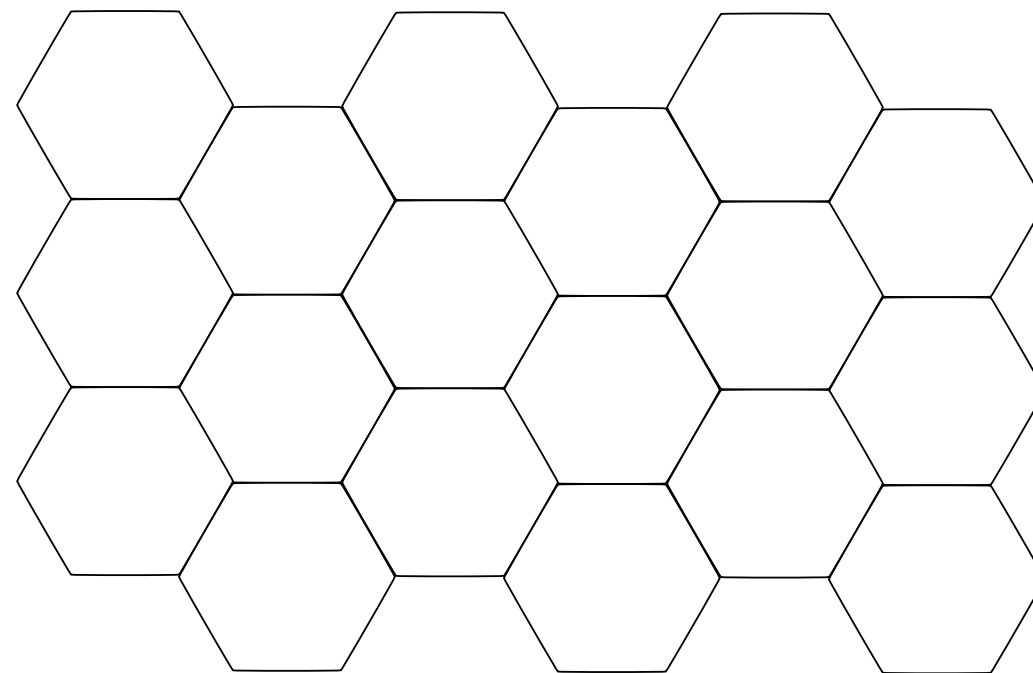


Exempel: Warlords



# Hexagonal värld och metrik

Hexrutor - brädspelarens bästa vän!



Ger bra avstånd och undviker många problem med kartesisk grid



# Taktisk AI

Planering av förflyttning

“Crash & turn”

Dijkstras algoritm

$A^*$

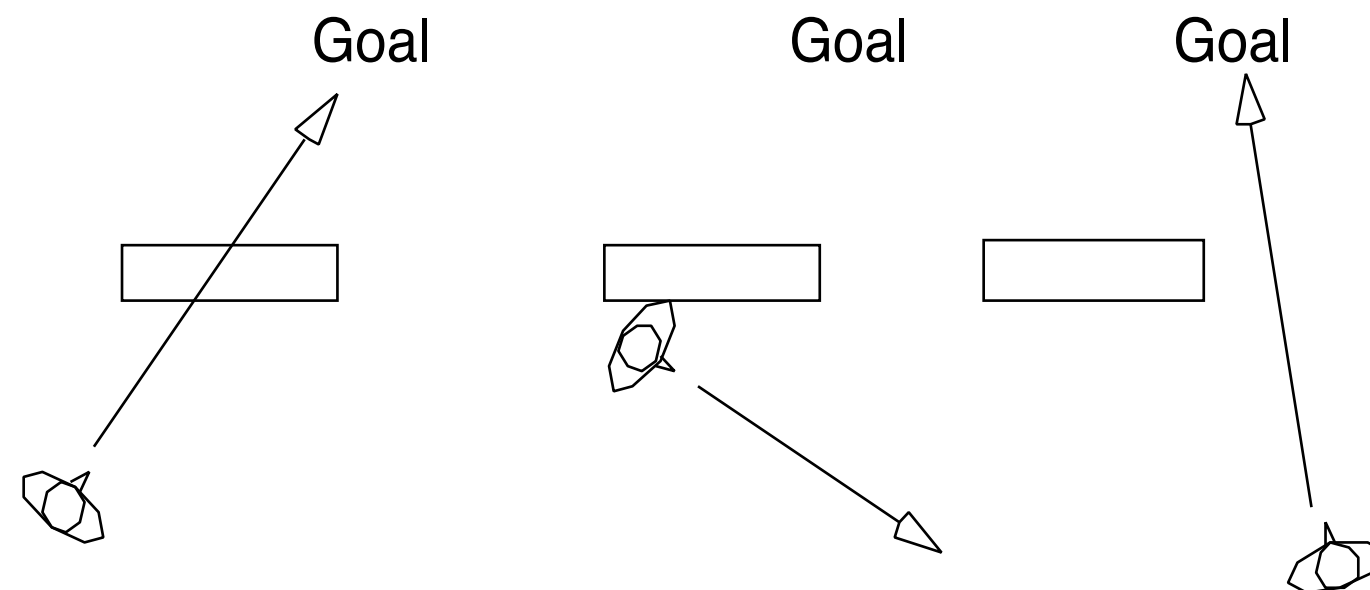
Omplanering när en väg blir oframkomlig

Är den optimala vägen rätt?



## “Crash & turn”

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

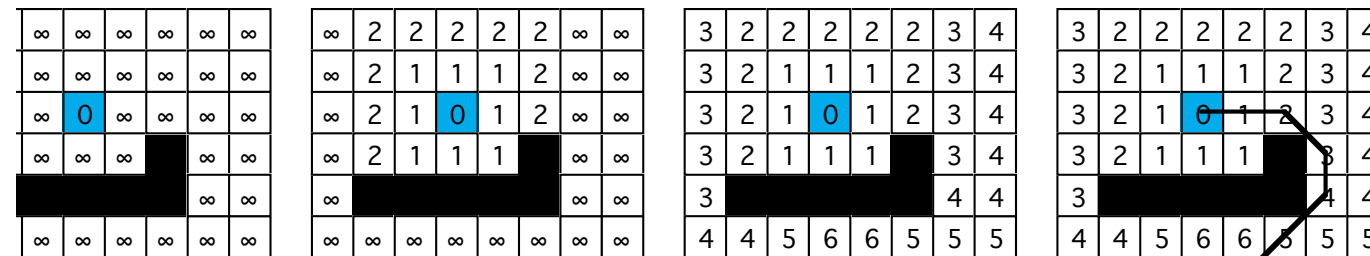


Fånigt? Planlöst? Det beror på vilket beteende som agenten bör ha!



## 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 hållet först

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



## Men A\* fuskar ju!

Optimal kortaste vägen, eller approximationen A\*,  
antar ofta att vi ser genom väggar eller har  
komplett information!

Crash & Turn är mer realistisk, men för korkad.

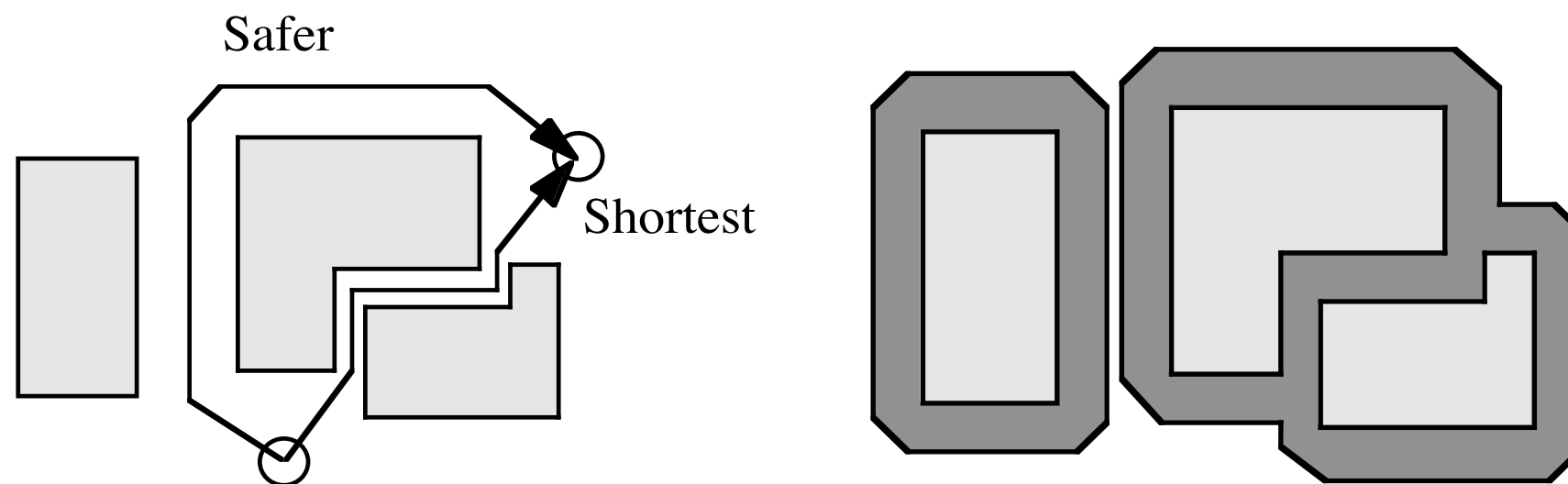
Crash & Turn med minne är närmare verkligheten.



## Säkraste vägen?

Prioritera vägar beroende på

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



Breda vägar kan hittas med matematisk morfologi -  
expansion av hindren





Information Coding / Computer Graphics, ISY, LiTH

# Sökning av tillståndsrymden

Minmax, Negamax

Metoder speciellt lämpade för deterministiska brädspel

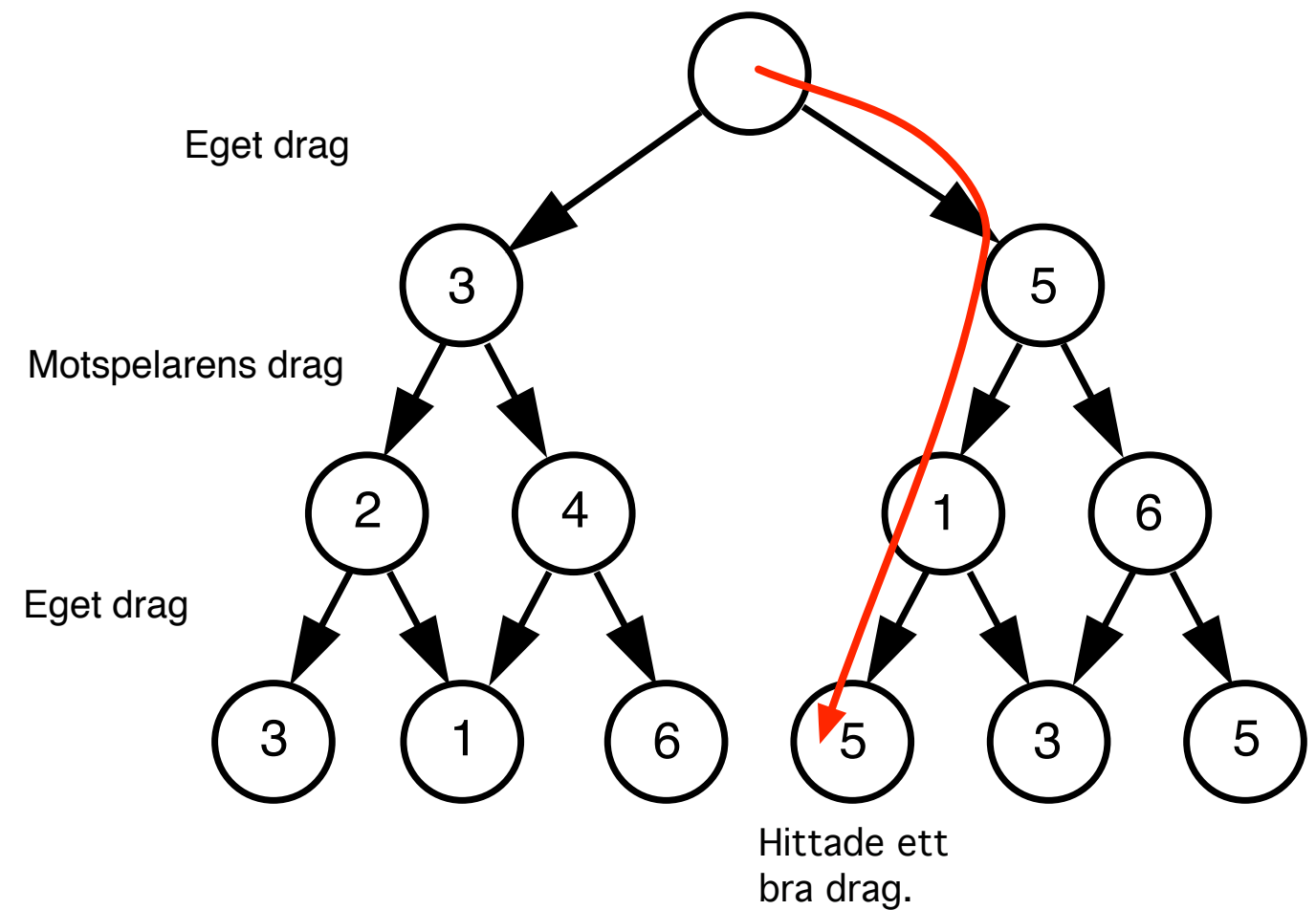
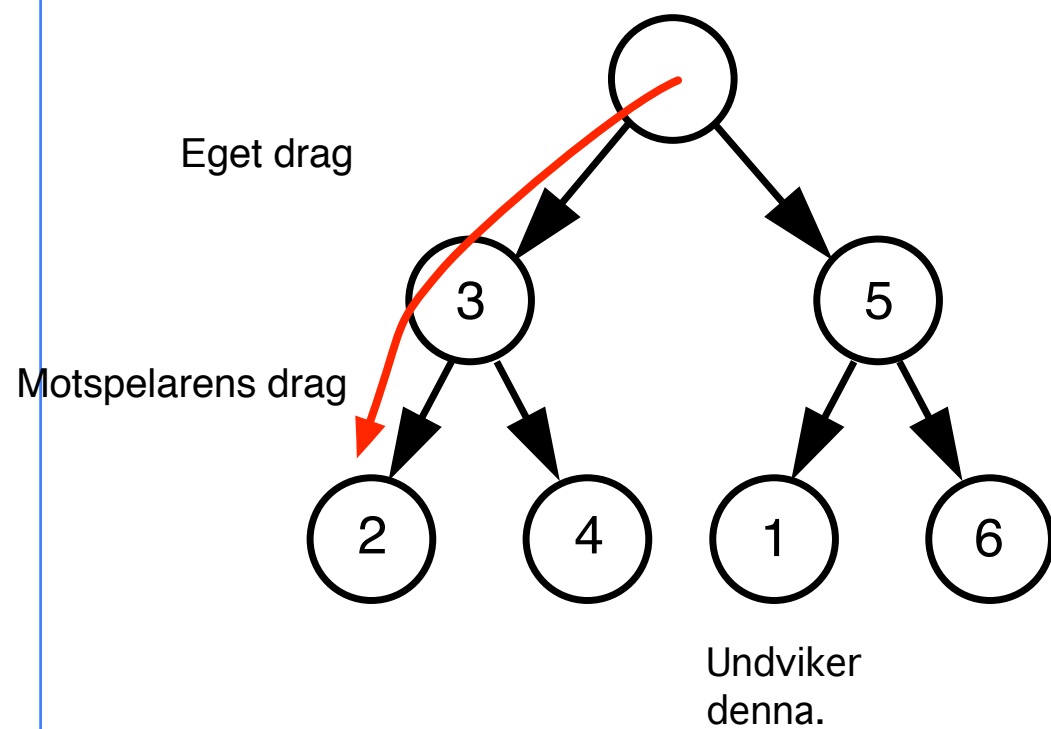
Sök den där mitt bästa drag är bäst och motståndarens  
bästa drag är sämst

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



# Minmax, exempel

Värdet nedan är hur bra det är för *mig*!



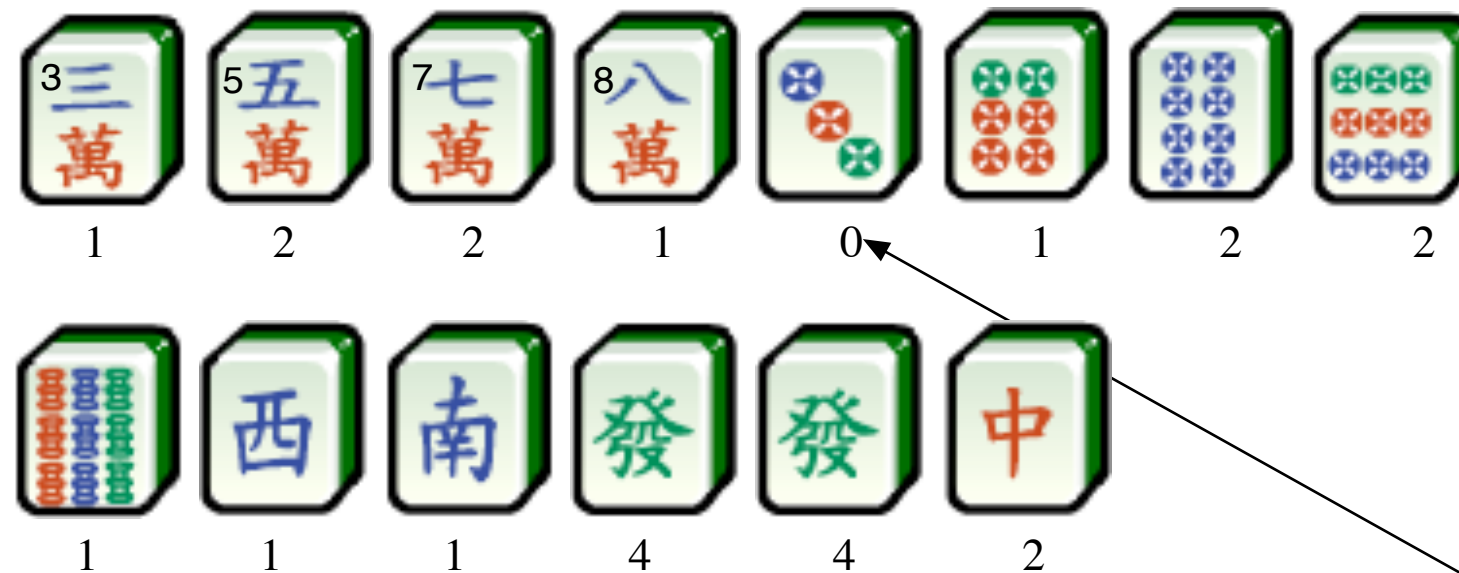


# Heuristisk utvärderingsfunktion

Lämpat för bl.a. icke-deterministiska brädspel

Gammalt favoritexempel:

(Klassiska reglerna!)



+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)

Släng den!