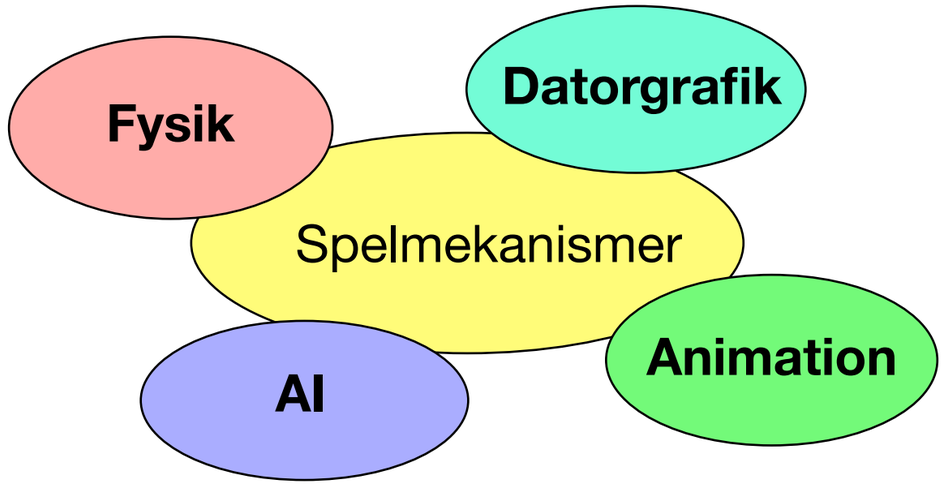


TSBK 03
Teknik för avancerade datorspel

Ingemar Ragnemalm, ISY



Labbarna

E-posthjälp såväl som i labbet

Möjlighet att göra om ALLA eller utvalda duggor på FÖRSTA reservlabben (8: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.)

Projekten

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

Spec inne till fredag 3/10.

Gör något intressant, men inte något alltför komplext!

Projektspecen

Kortfattad!

Titel + Vem

Sammanfatta vad som skall göras

Lista av skall- och börkrav.

Vari består milstolpen?

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?



Föreläsning 10: Spel-AI

Beteenden & beslut
Världsrepresentation
Metriker
Influence maps
Flocking

Föreläsningsfrågor

1. Hur kan en FBM användas för att modellera ett beteende?

2. Vad är fördelen med GOAP över FBM?

3. Vad gör Wave Function Collapse? Vad är "kollaps"?

4. Varför heter det "minmax"? Vad är det som minimeras och maximeras?

5. Vad är fördelen med hexagonal karta över kartesisk?

6. Varför lågpassfiltreras en infuence map?

7. Vad är de tre reglerna för 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

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, beteende
Vad skall agenten utföra? Förflyttningar mm.

Fusk
Bör agenten tillåtas fuska?

Vanliga metoder

- Finita tillståndsmaskiner (FSM)
- 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:

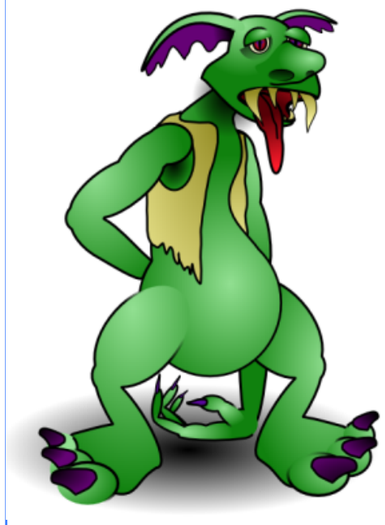
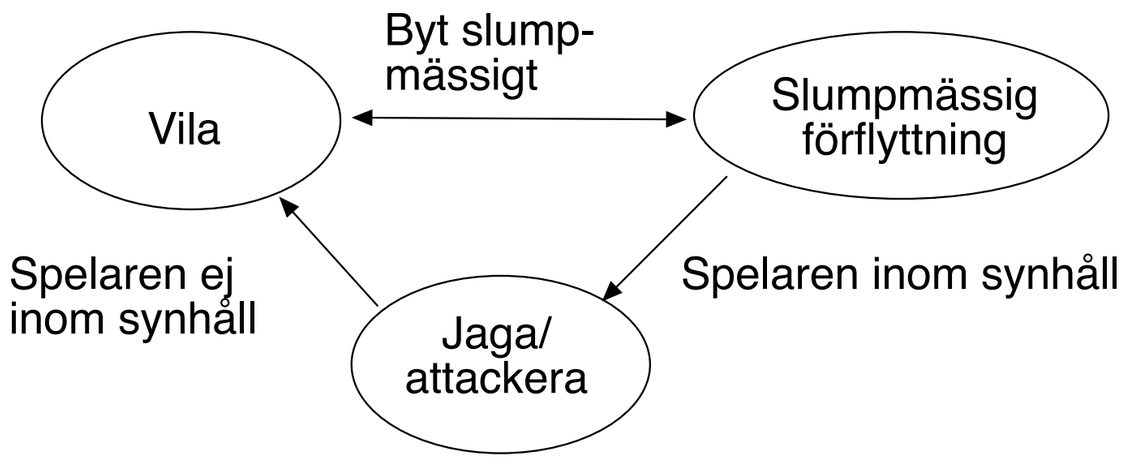
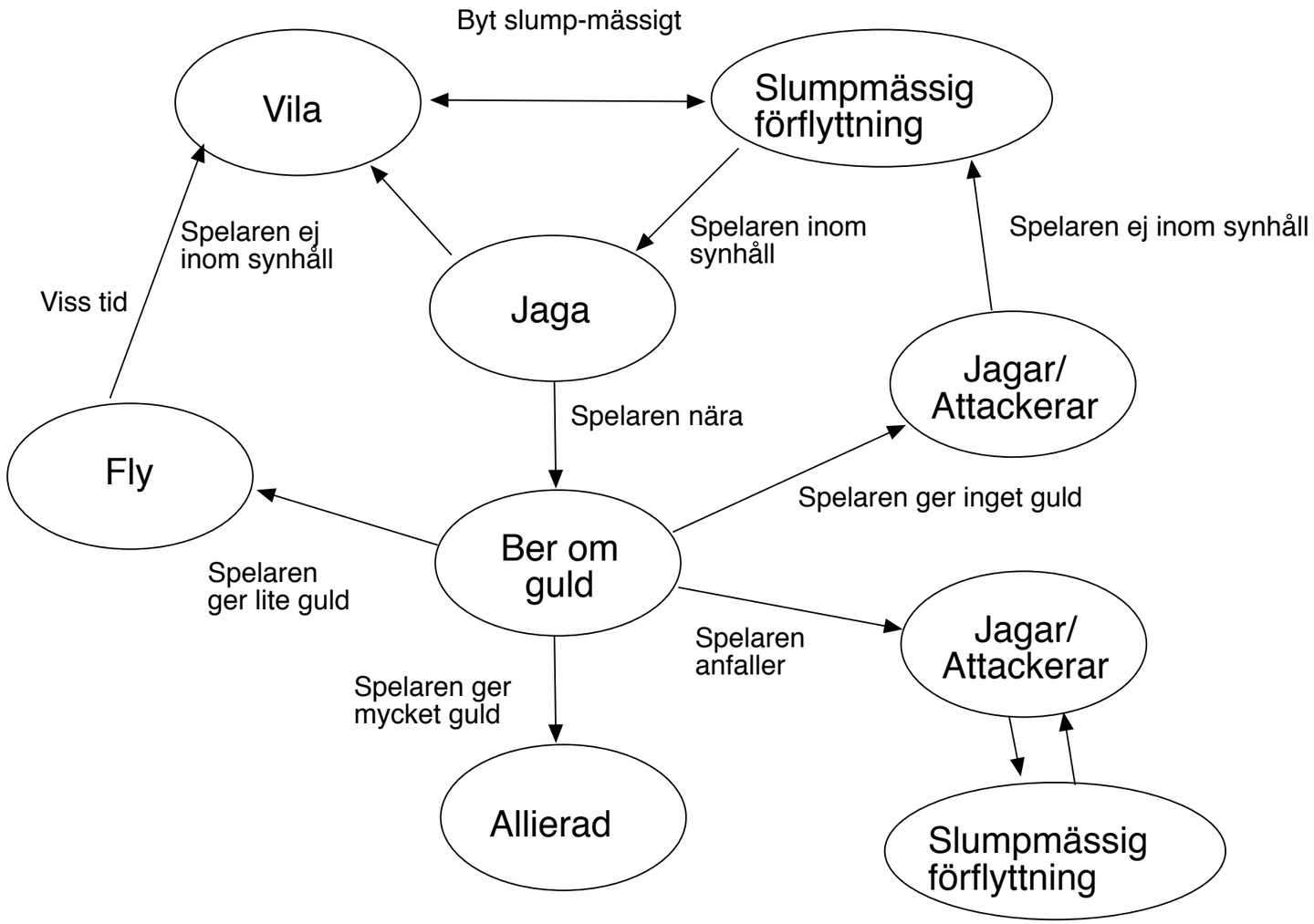


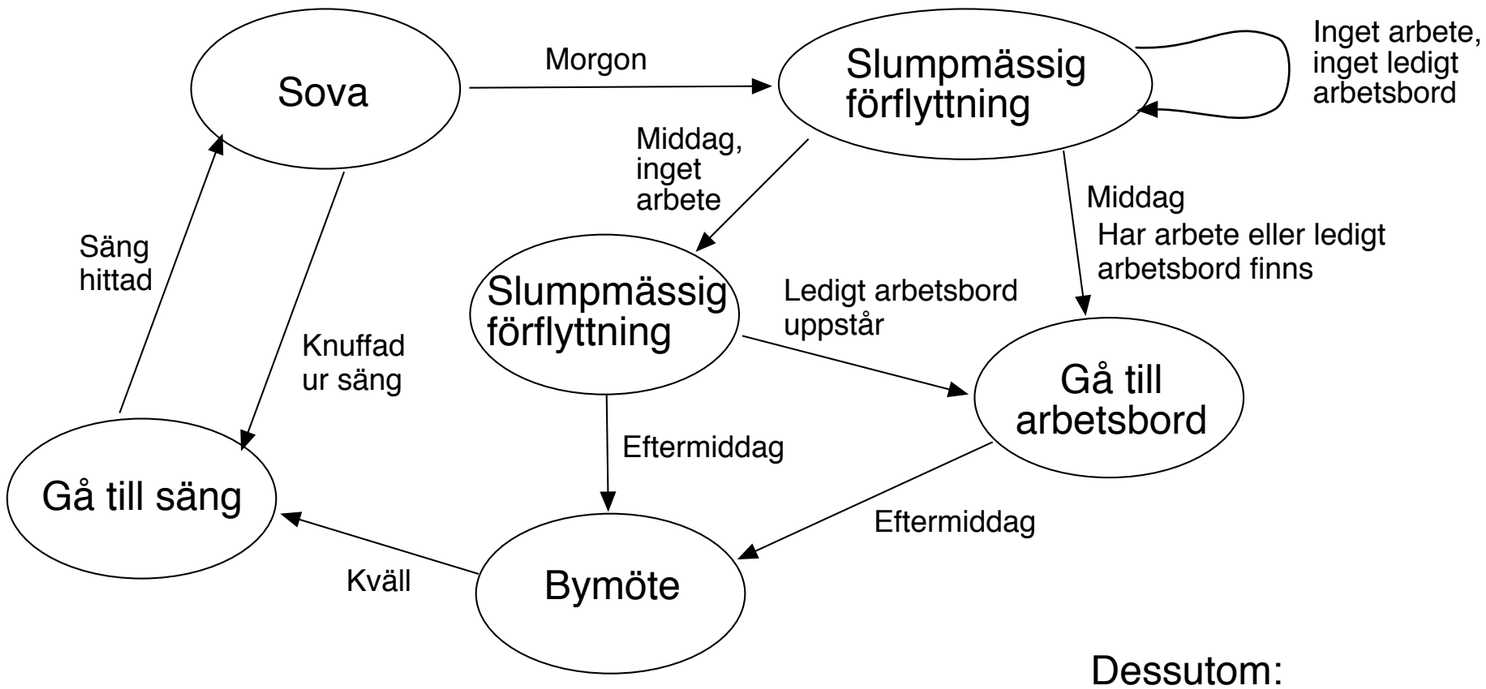
Bild från pdclipart.org



Något mer avancerat exempel



Villager



Vissa beteenden under bymöte:
Prata med andra bybor.

Dessutom:
- Love mode
- Village hero-gåvor

Regelsystem

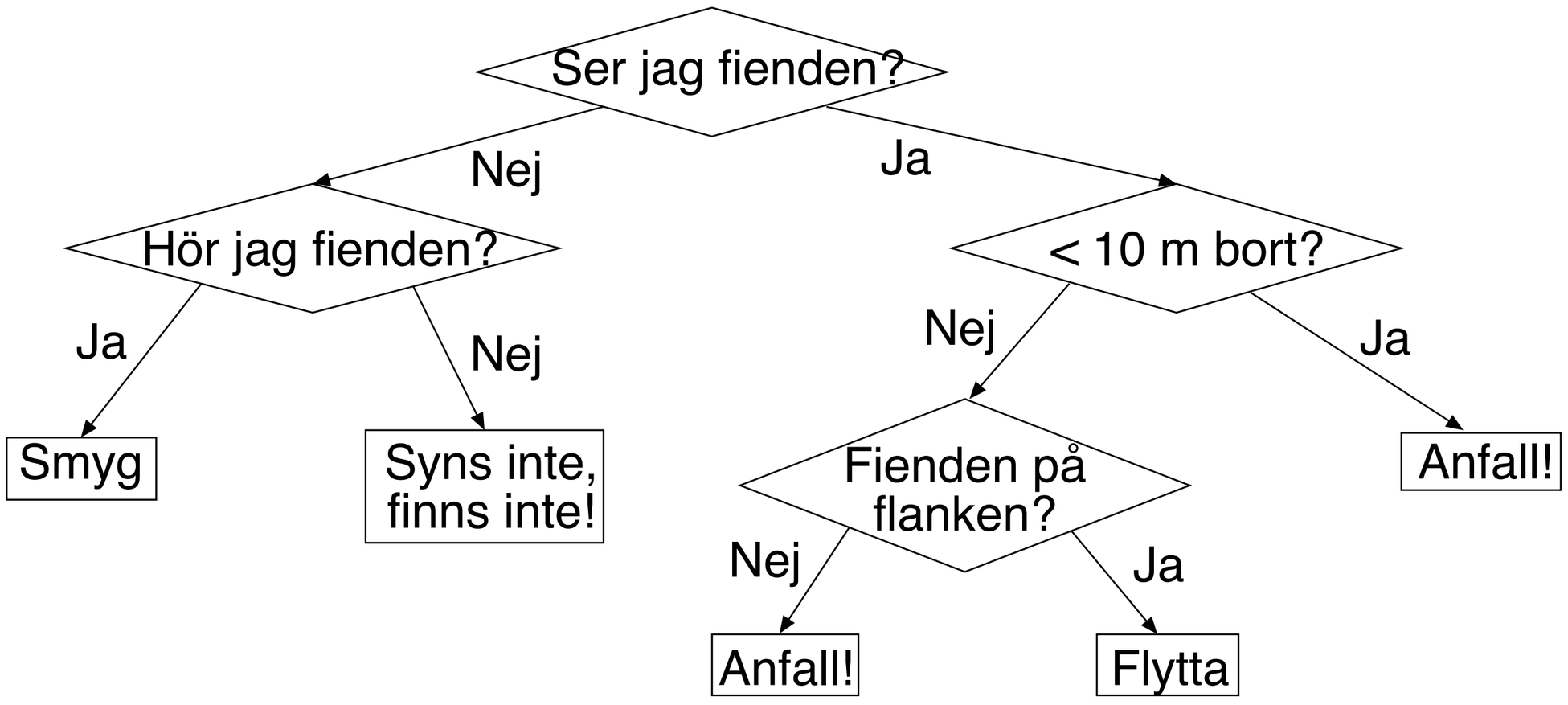
Enkla regelsystem kan representera många action-AI

- 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.

Beslutsträd

Regelsystem bildar ofta en struktur



Vissa beslut kan tillföras slump.

En del information kan vara en skala, vilken motiverar slump och inte bara ett gränsvärde.

Kan kombineras med FSM.

Goal Oriented Behavior

Målbaserat beteende

En agent kan ha *mål*.

Styr beteendet med dessa.

Ge dem olika prioritering beroende på situation.

Exempel (Millington): Hunger och sömn

Goal: Eat = 4
Goal: Sleep = 3
Action: Get-Raw-Food (Eat – 3)
Action: Get-Snack (Eat – 2)
Action: Sleep-In-Bed (Sleep – 4)
Action: Sleep-On-Sofa (Sleep – 2)

Välj efter vad som är mest kritiskt och vad som är lättillgängligt.

Behov kan påverka varandra

Törst -> Drick -> Behöver gå på toa

Springa för att undvika fara -> Trött -> Behöver vila

Goal Oriented Action Planning

En modell för att hantera kedjan av händelser för att nå ett mål.

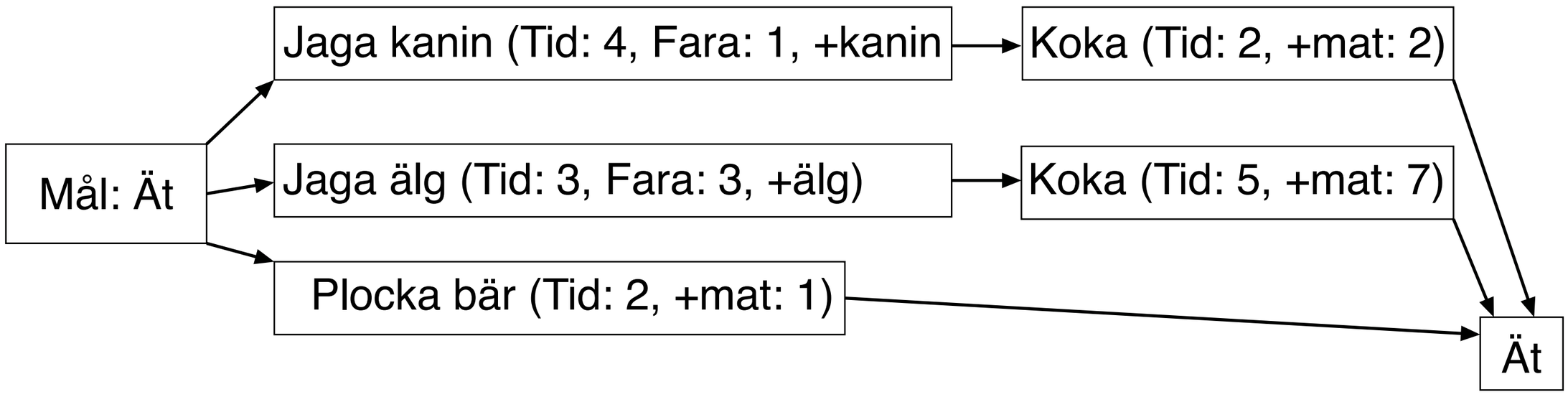
Skapa träd av möjliga handlingar som kan nå målet.

Trädet beror av situationen, vad som är möjligt.

Sök trädet för bästa lösningen på problemet.

GOAP

Enkelt exempel



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?

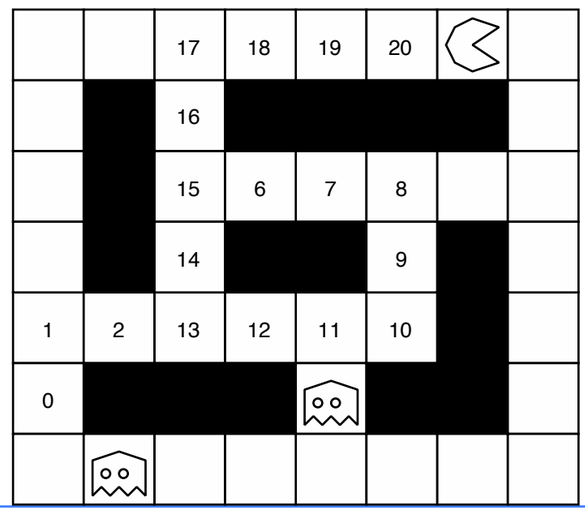
Världrepresentation

Exempel:

Pac-Man-liknande spel

Världen är en grid, representeras med array

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

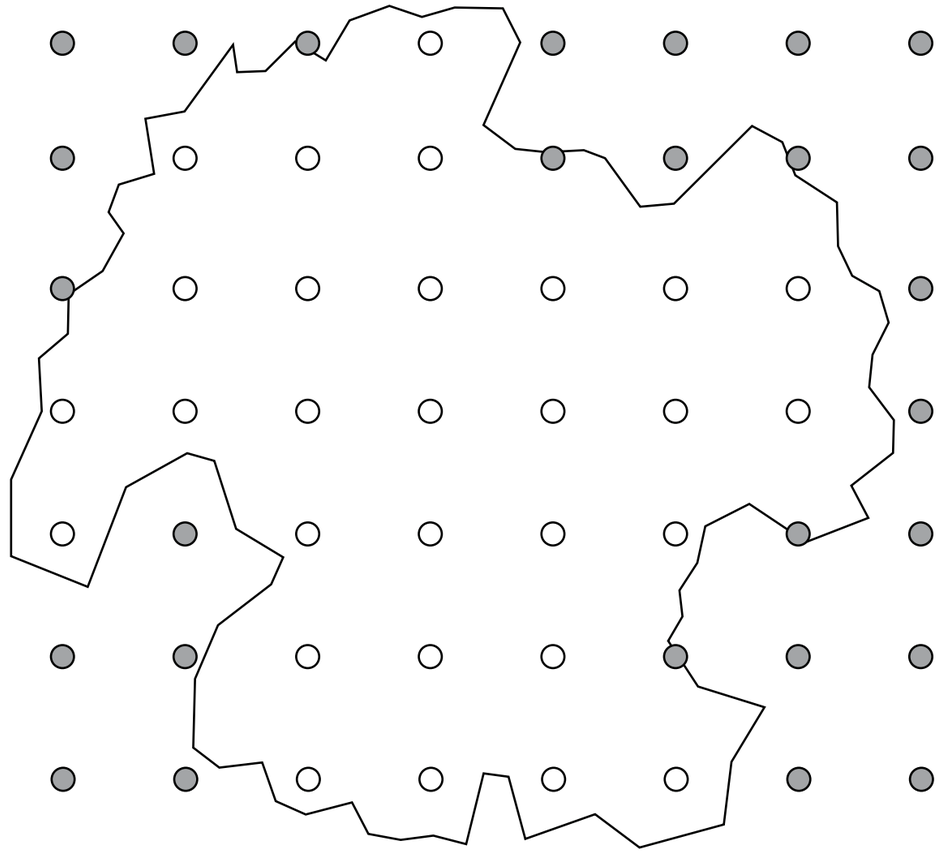


Världrepresentation

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 (Tidiga versioner)

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

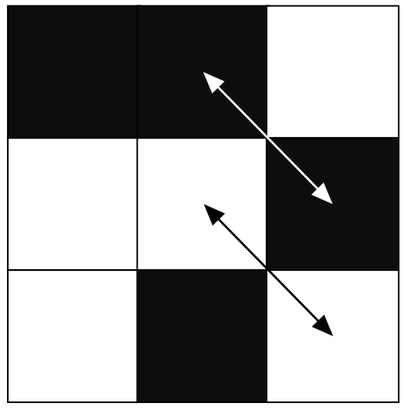
Chamfer 1-$\sqrt{2}$

Konnektivitet

Vad är granne till vad?

Relaterat till metriker

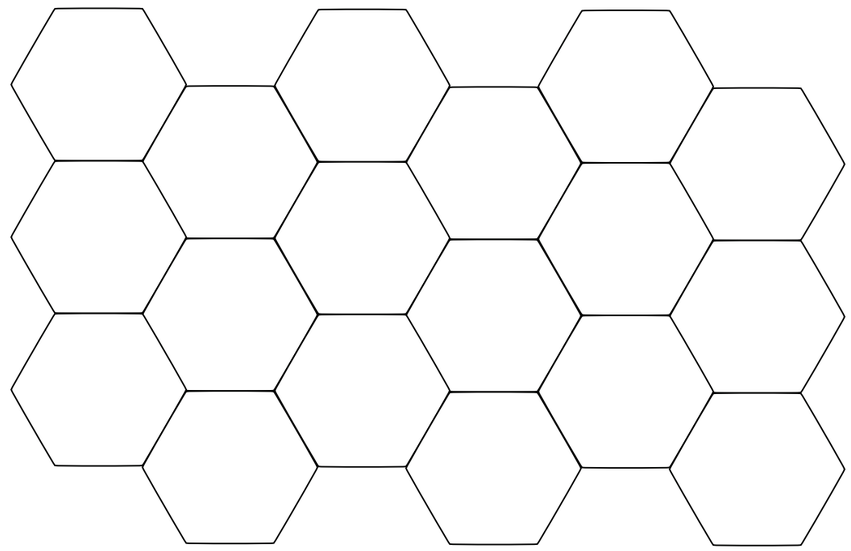
d4-konnektivitet för en domän ger d8 i den andra!



Exempel: Warlords

Hexagonal värld och metrik

Hexrutor - brädspelarens bästa vän!



Ger bra avstånd och undviker många problem med kartesisk grid

Generering av kartor

Hur man genererar kartor/nivåer procedurellt

Vi kan redan: Generering av höjdkartor med FBM

Hur fyller vi kartan med olika egenskaper?

Biomer: Generera med låga frekvenser. Tröskla fram biomer.

Kartor med beroenden

Vad för objekt passar nära varandra?

Wave Function Collapse

Mycket förbryllande namn, syftar på kvantteori.

En metod för att iterativt skapa kartor där olika typer av celler, biomer eller liknande placeras för att undvika olämpliga grannar.

Collapse = bestämd, fixerad, konvergerad

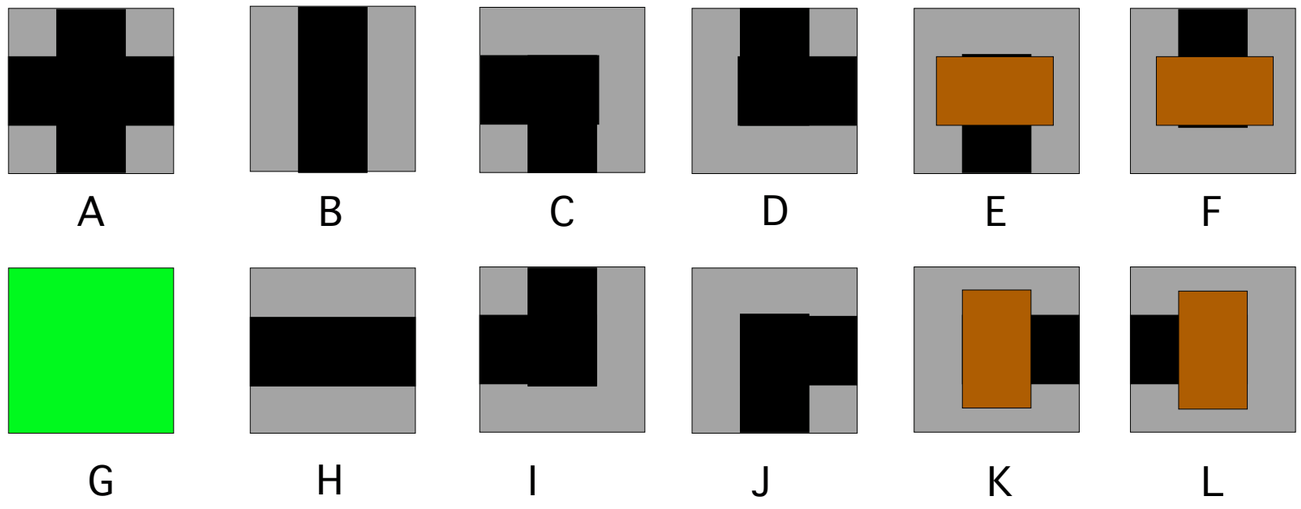
Beroende på grannarna kan vi bestämma vilka värden som är tillåtna.

När bara ett är tillåtet "kollapsar" värdet och blir bestämt.

Varje gång en "kollapsar" begränsas valen för grannar så de också kollapsar eller kommer närmare att kollapsa.

Enkelt exempel

Vägar, gräs, Hus



Start: Allt är tillåtet

Alla riktningar har en lista av alla.

ABCDEFGHIJKL

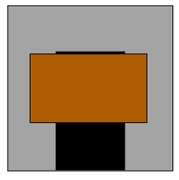
ABCDEFGHIJKL ? ABCDEFGHIJKL

ABCDEFGHIJKL

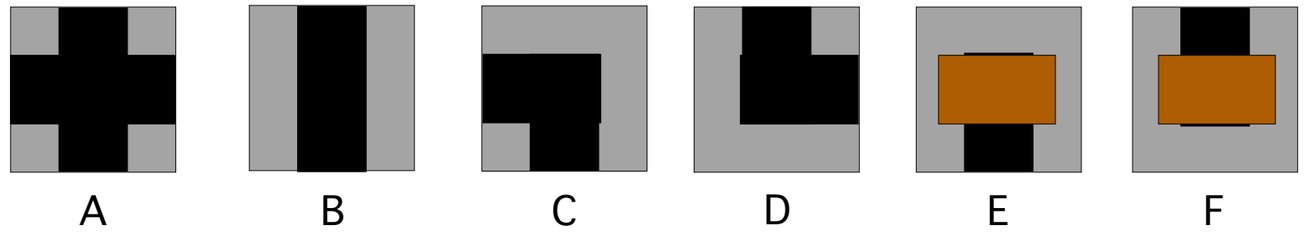
Välj en cell slumpmässigt, sätt den

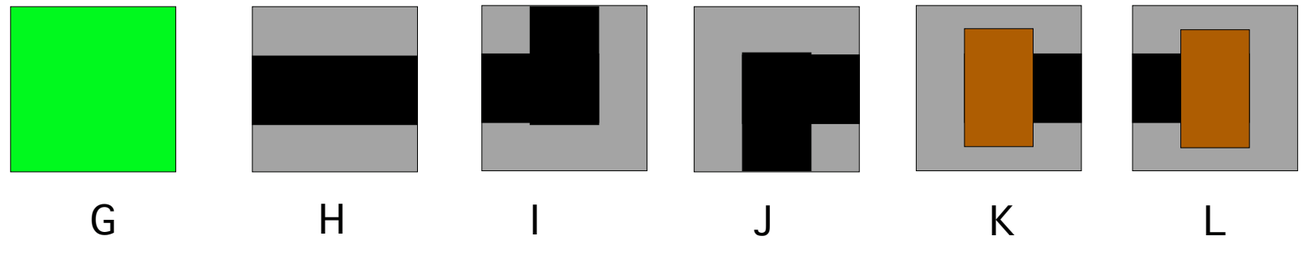
Grannlistorna begränsas, kan påverka i flera steg

DFGHIKL

BEFGIL  BDEFGJK

ABDI

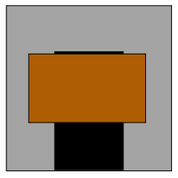




Möjligheterna minskar

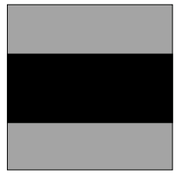
Grannlistorna reduceras till gemensamma element

DFGHIKL

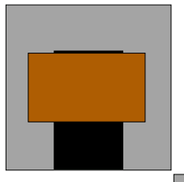
BEFGIL  BDEFGJK

ABDI

DFGHIKL

ADHJK  ACHIL

CEHJKL



Snitt(DFGHIKL, BDEFGJK)
= DFGK

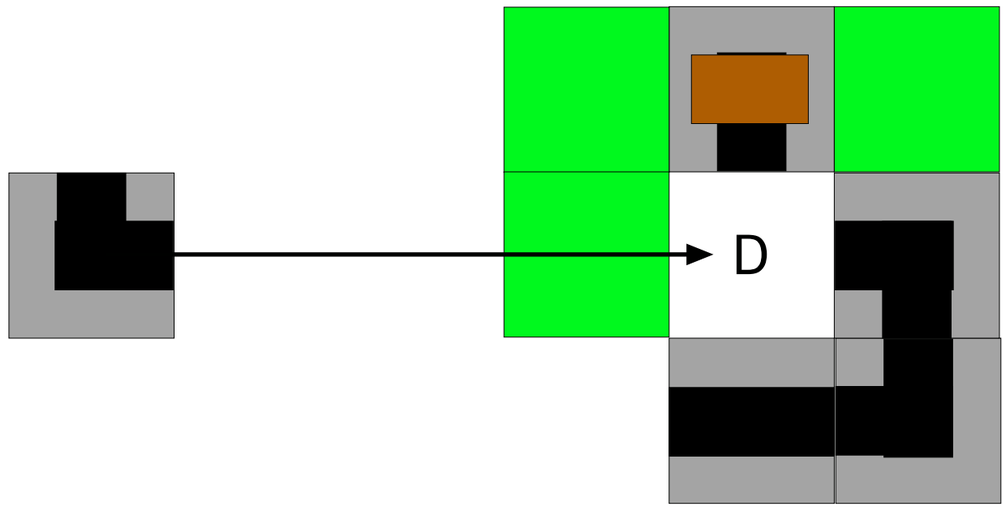
Snitt(ABDI, ADHJK)
= AD



Upprepa tills kartan är full

eller konflikter uppstår. Backa eller börja om.

Regel: Fixera celler med minst möjligheter!



Taktisk AI

Planering av förflyttning

“Crash & turn”

Dijkstras algoritm

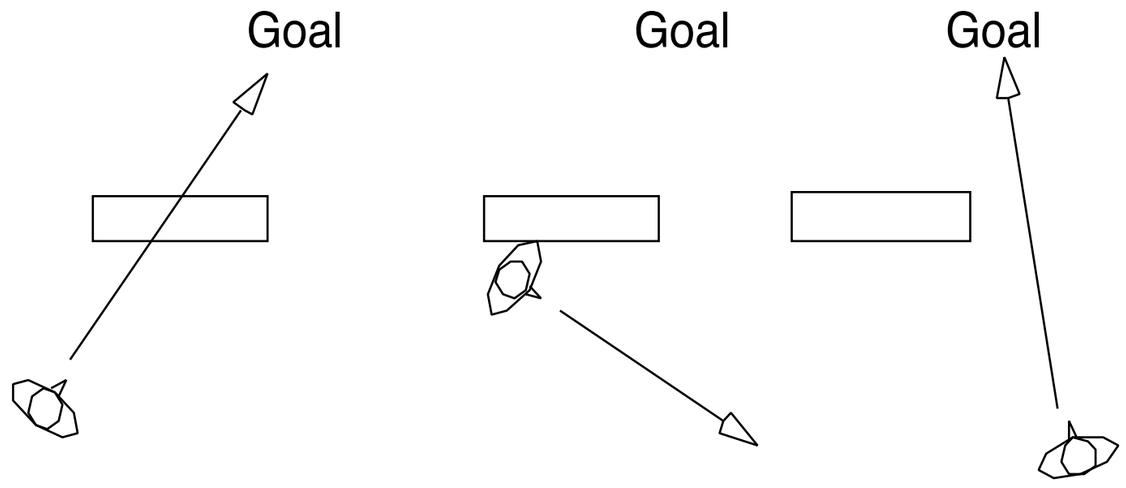
A*

Omplanering när en väg blir oframkomlig

Är den optimala vägen rätt?

“Crash & turn”

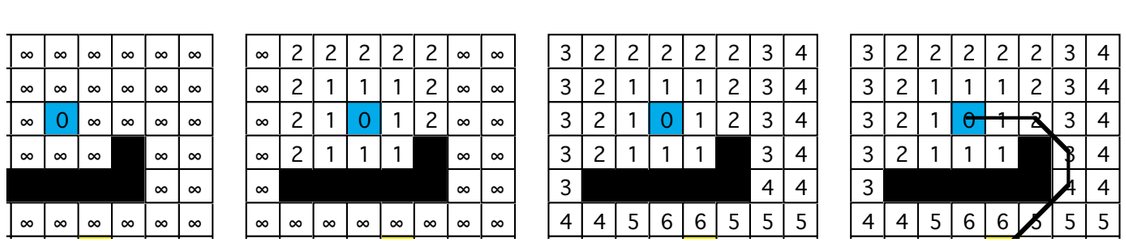
- Gå mot målet tills nånting tar emot
- Gå i slumpmässig riktning
- Upprepa



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!

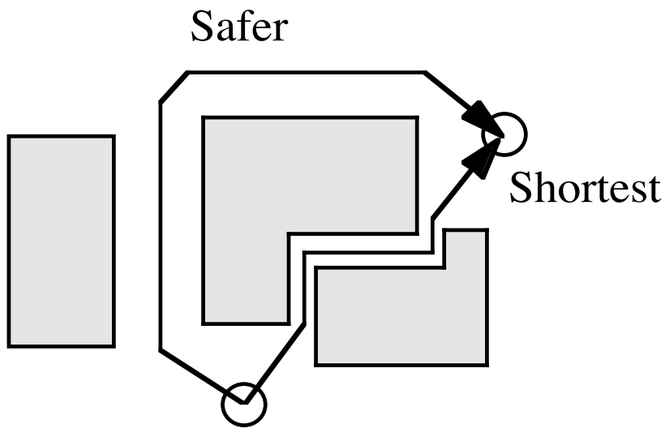
Optimal kortaste vägen, eller approximationen A*,
antar ofta att vi ser genom väggar elller har
komplett information!

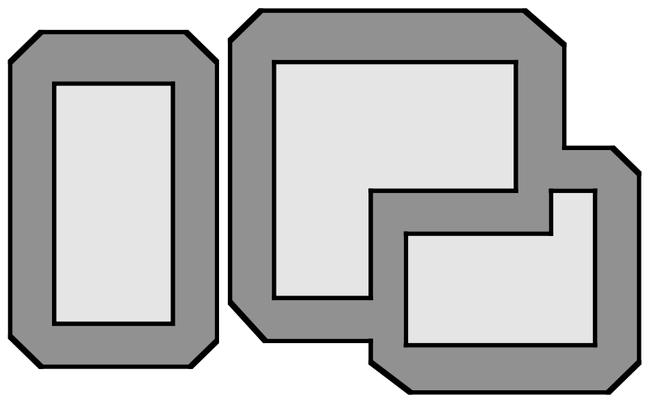
Crash & Turn är mer realistisk, men för korkad.

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

Sökning av tillståndsrymden

Minmax, Negamax

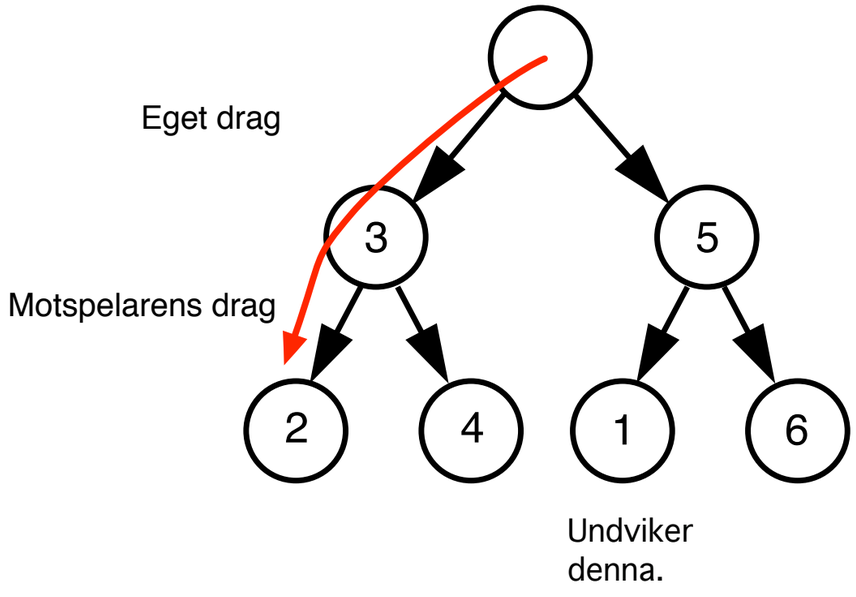
Metoder speciellt lämpade för deterministiska brädspel

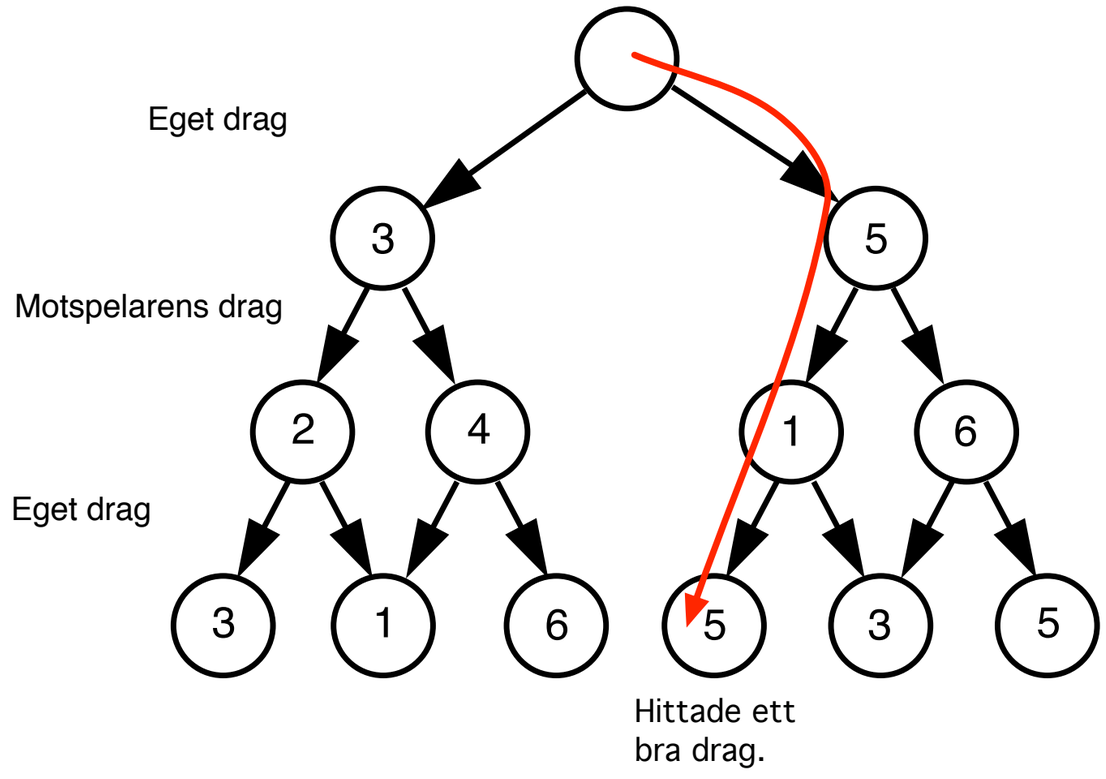
Sök den där mitt bästa drag är bäst och motståndarens bästa drag är sämst

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

Minmax, exempel

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

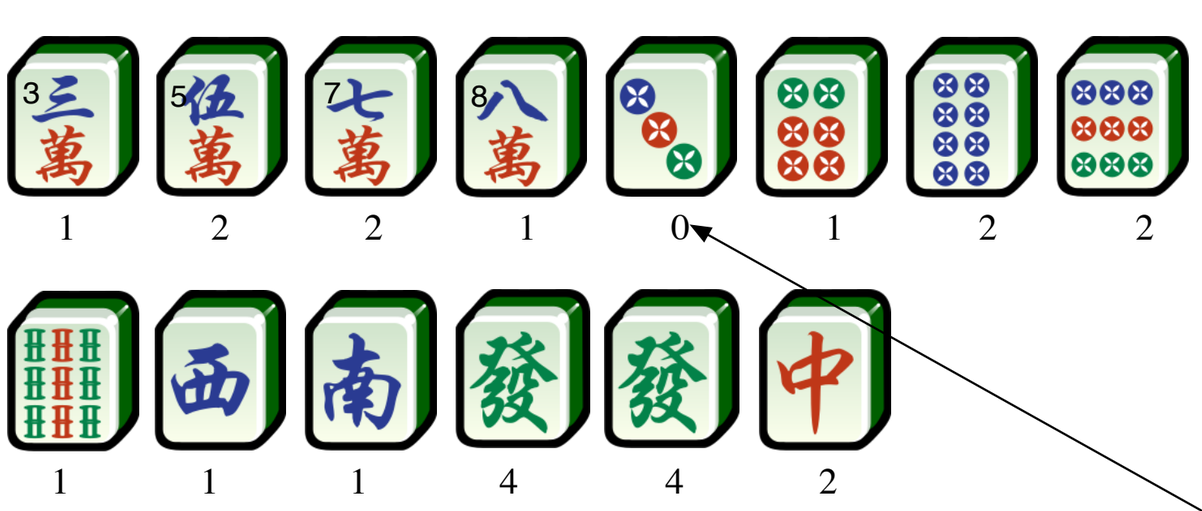




Heuristisk utvärderingsfunktion

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)

Brickor från Wikipedia