

Stifinder

V. Allan Ebdrup

Heuristik til at finde sti for firkantet
objekt mellem firkantede
forhindringer

Motivation

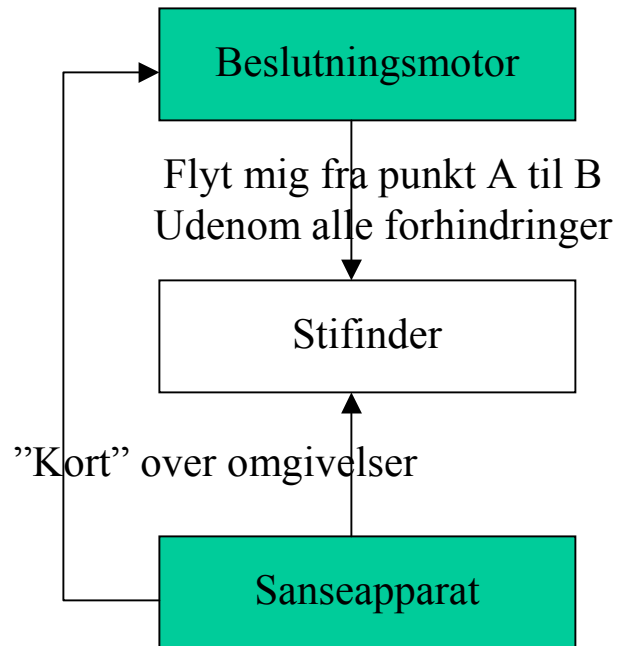


- dotNet Terrarium
<http://www.getdotnet.com/terrarium/>
- Rovdyr
 - Jagte andre dyr
 - Løbe væk fra farligere dyr
 - Avanceret, fx jage i flok
- Planteædere
 - Finde planter
 - Løbe væk fra angreb
 - Avanceret, fx forsvare i flok
- Krav: Finde sti rundt om planter og andre dyr.

Dagsorden

- Problemområde
- Sanseapparatet
- Simplifikationer i stifinder
heuristikken
- Kollisionsdetektion
- Gennemgang af
heuristikken
 - Find nærmeste forhindring
 - Find sti udenom forhindring,
med og uden overlap mellem
forhindringer
 - Cykler
- Komplexitet
- Konklusion
- Forbedringer &
udvidelser
- Opsamling &
afrundning

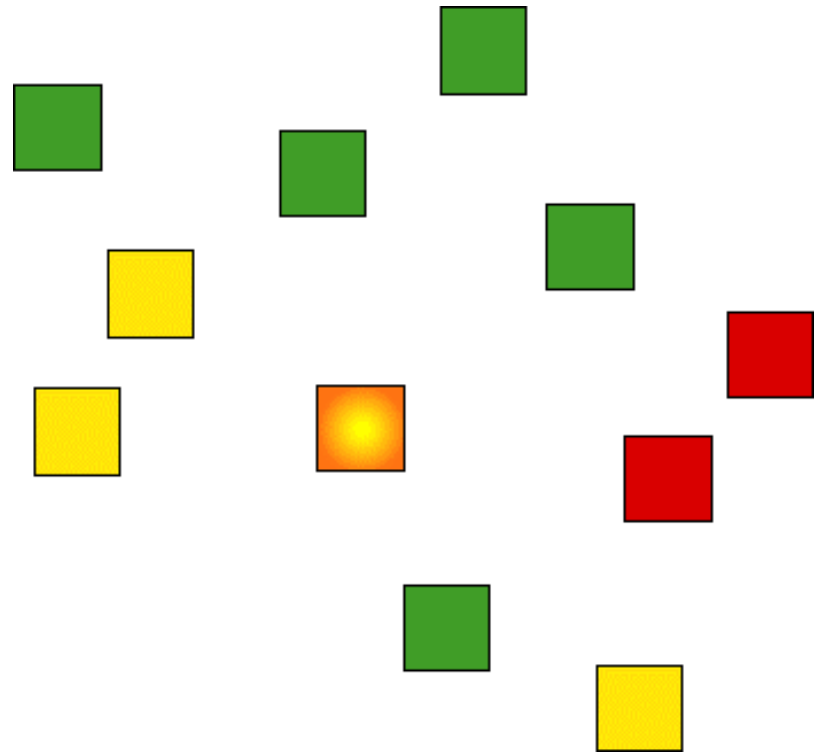
Problemområde



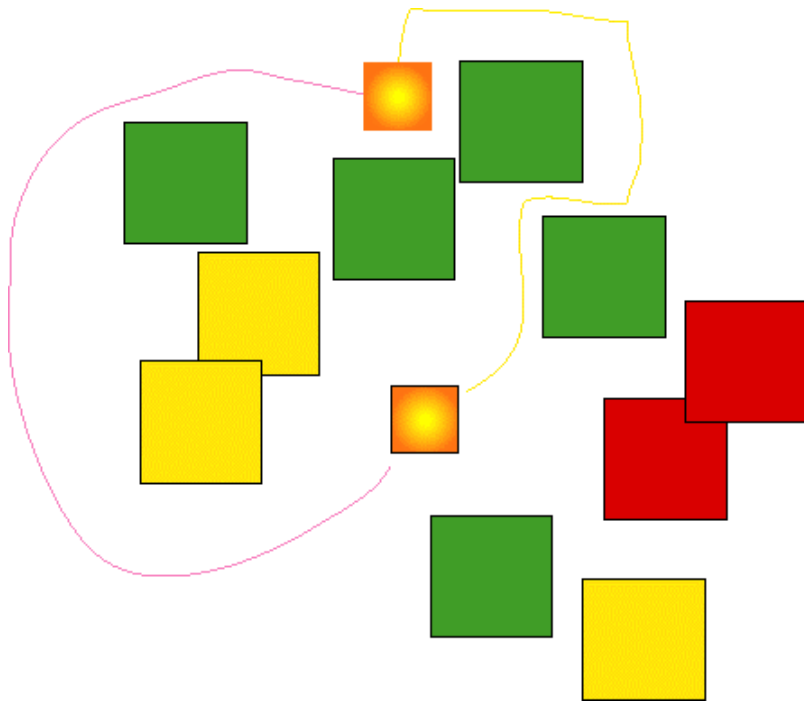
- Beslutningsmotor udelades
- Sanseapparatet er implementeret i dotNet Terrarium
- Der kigges altså kun på stifinder delen

Sanseapparat

- Man kan "se igennem" alt, man kender området i forvejen.
- 360 grader syn
- Alle objekter optager plads beskrevet ved en firkant.
- Firkanterne kan ikke drejes
- Ingen lugte eller lyde
- Begrænset følelse, man "mærker" når man løber ind i noget.



Kollisionsdetektion

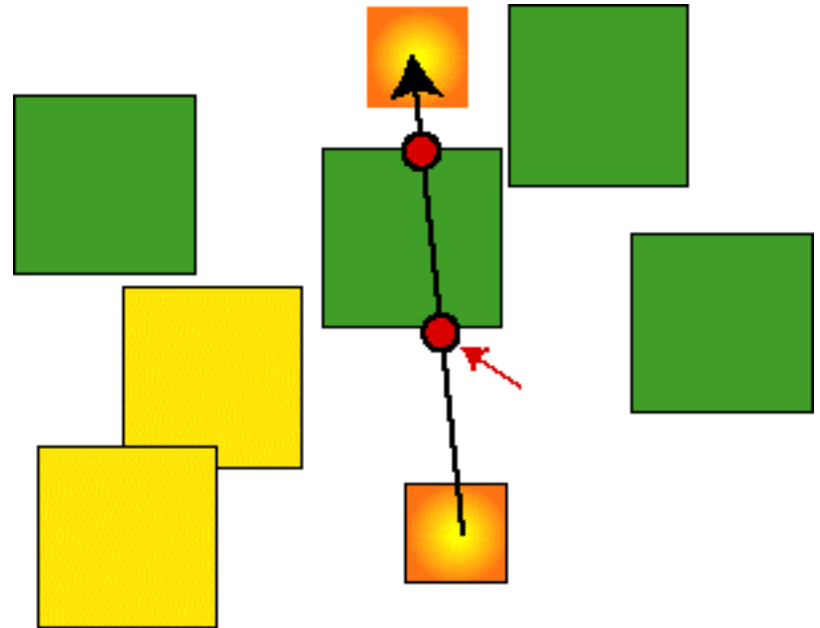


- Alle forhindringer forstørres og stien reduceres til en linie, dette giver følgende egenskaber:
 1. Kollisionsdetektion mellem objektet og en forhindring reduceres til at checke om objektets midte ligger indenfor den forstørrede forhindrings areal
 2. Kollisionsdetektion kan bruges til at finde forhindringer der overlapper, hvilket betyder at objektet ikke kan komme imellem dem.
 3. Forhindringer der ikke allerede er "indeni" hinanden vil ikke komme "indeni" hinanden.

Stifinder heuristikken

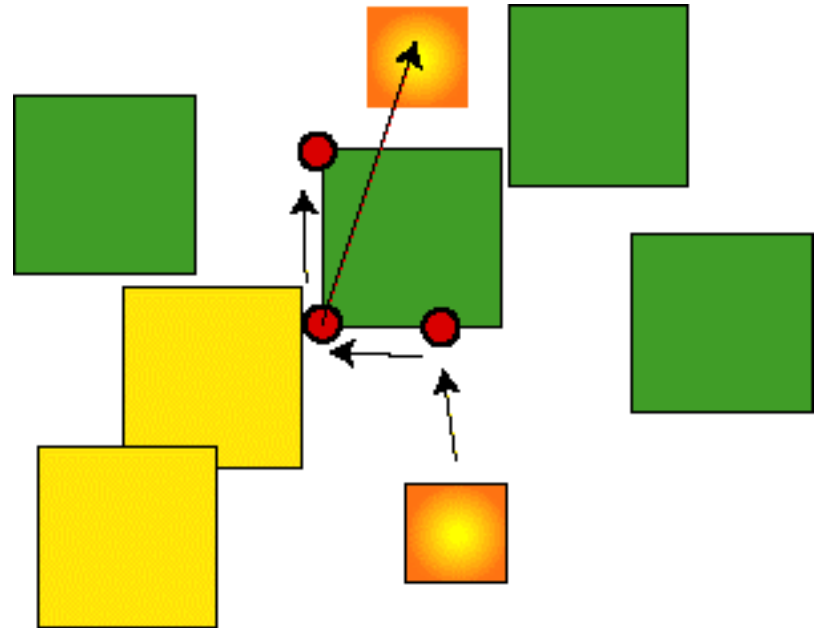
- find nærmeste forhindring

- Find alle lineskæringer med linien fra A til B
- Tag fat i det tætteste skæringspunkt og find vej udenom den forhindring der gav dette skæringspunkt



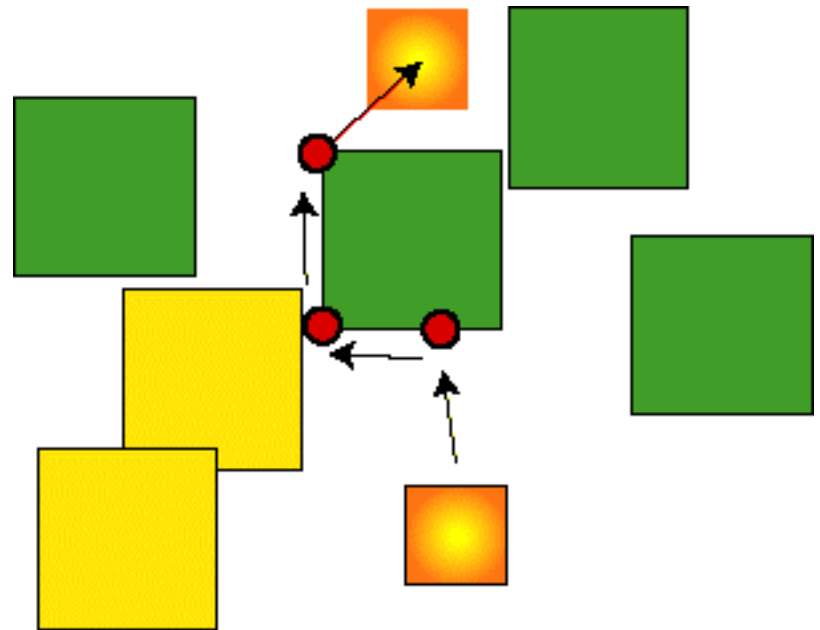
Find Sti Udenom Forhindring

- Følg forhindrings yderkant mod uret til næste punkt
- Tegn linie til B fra nyt punkt, hvis stien stadig går gennem vores forhindring følges kanten rundt på forhindringen til nyt punkt



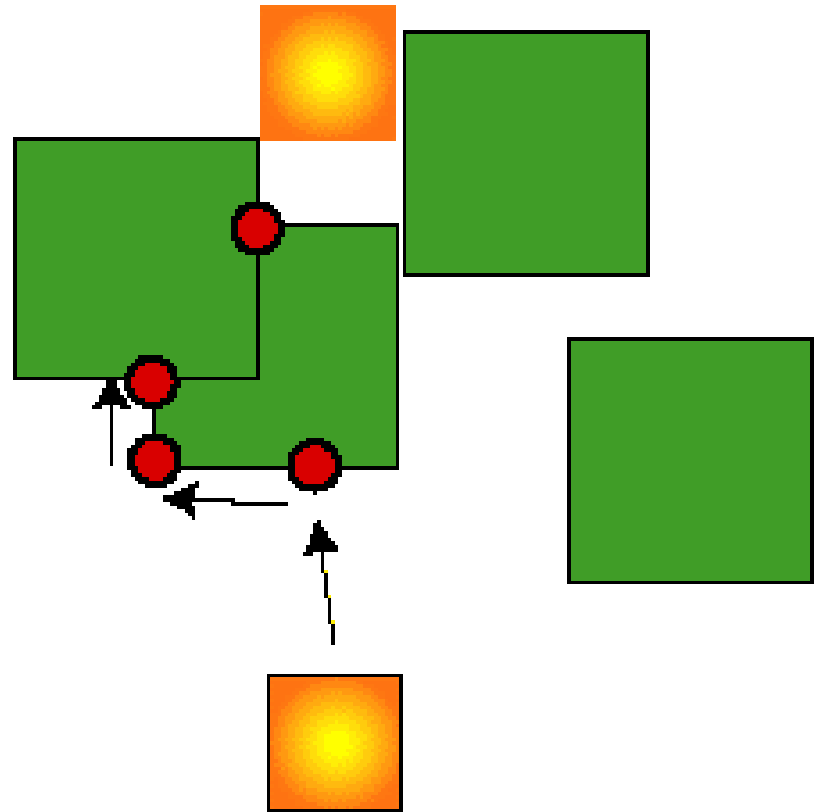
Find Sti Udenom Forhindring

- Hvis linien fra det seneste punkt til B ikke går gennem vores aktuelle forhindring kaldes stifinder algoritmen så vi finder en sti fra det seneste punkt til B, indtil der er fri passage til B



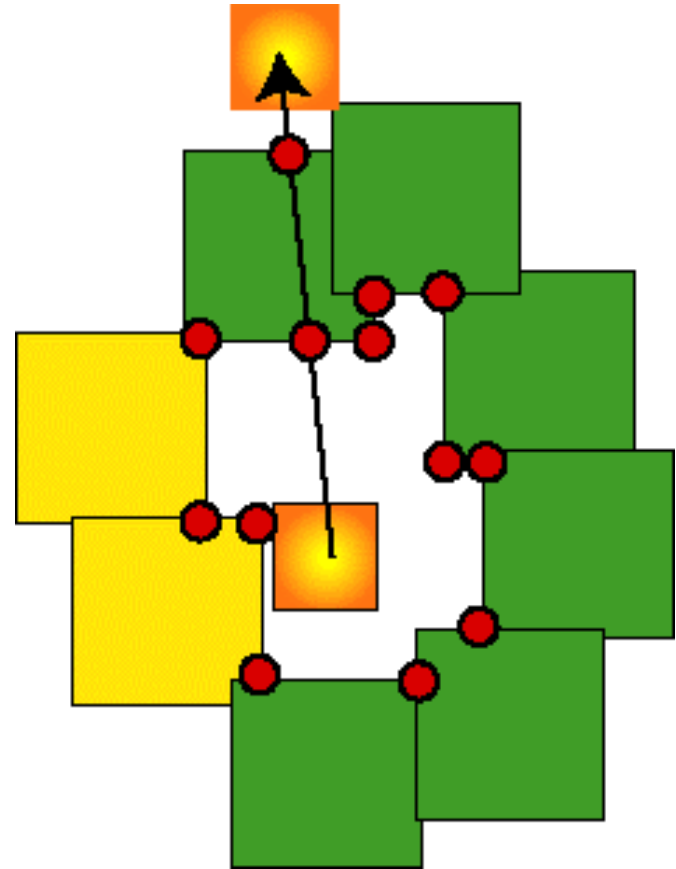
Overlap Mellem Forhindringer

- Hvis forhindringen vi er på vej udenom har overlap med en anden forhindring udregnes alle skæringspunkter med den anden forhindring
- Når vi skal finde det næste punkt på stien ”stopper” vi ved sådanne overlaps punkter og finder i stedet vej udenom den nye forhindring.



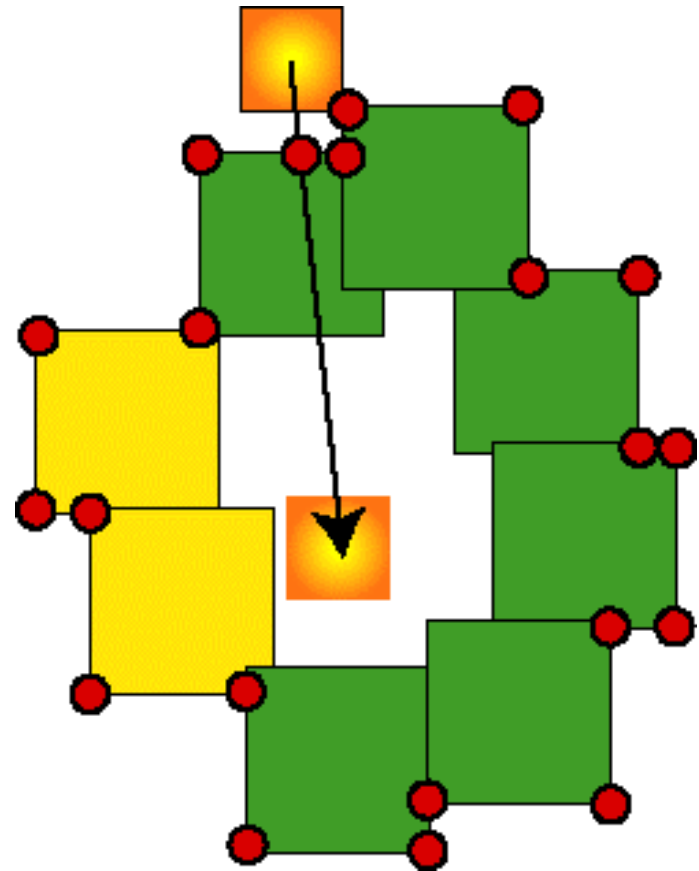
Cykler

- Der findes 2 slags cykler
 1. Cykler der opstår fordi vi er "spærret inde"



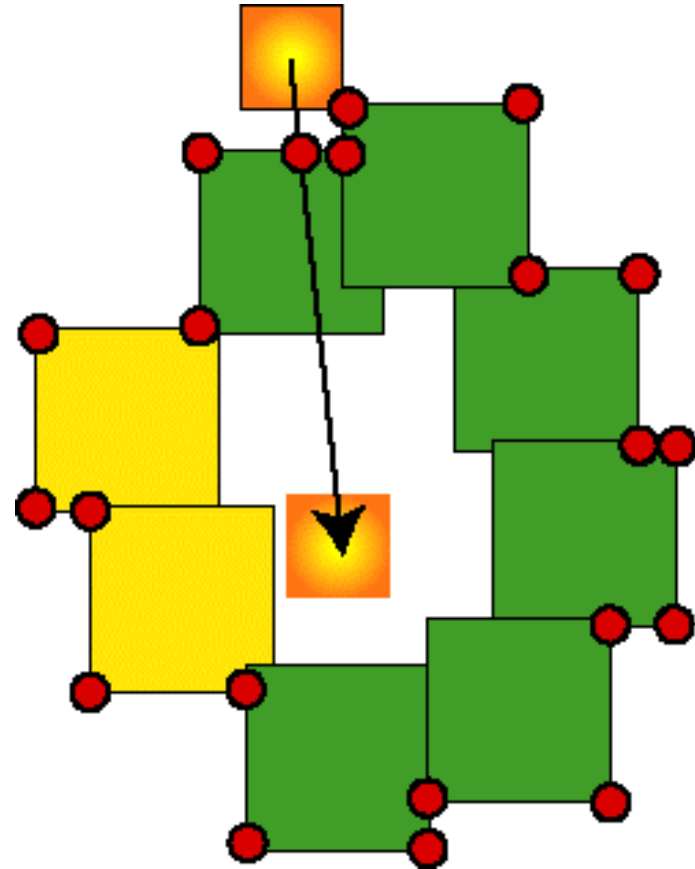
Cykler

2. Cykler der opstår fordi målet er ”spærret inde”
- Cykler kan opdages ved at markere skæringspunkter man allerede har besøgt
 - Problemet bør analyseres igen af beslutningsapparatet



Kompleksitet

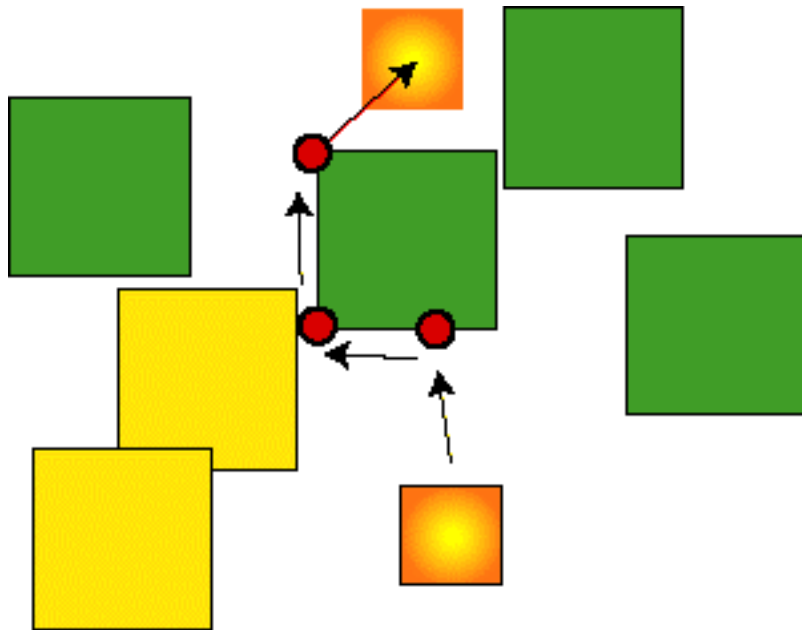
- Find tætteste forhindring: N
Vi kan risikere at skulle finde tætteste forhindring N gange:
 $N * N: O(N^2)$
- Find udenom forhindring
 - Find overlap mellem to forhindringer:
 $O(N^2)$
 - Find overlap mellem alle forhindringer:
 $0+1+2+3+4+\dots+(N-1): O(N^2)$
 - Worst case antal overlap:
 $0+2+4+6+8+\dots+2(N-1): O(N^2)$
 - Vi besøger hvert overlap maksimalt 2 gange og hvert hjørne af en forhindring maksimalt 2 gange.
- I alt $O(N^2)$ tid og $O(N^2)$ plads.



Konklusion

- Stifinder algoritmen - finder en sti for et firkantet objekt gennem en mængde firkantede forhindringer i $O(N^2)$ tid og $O(N^2)$ plads

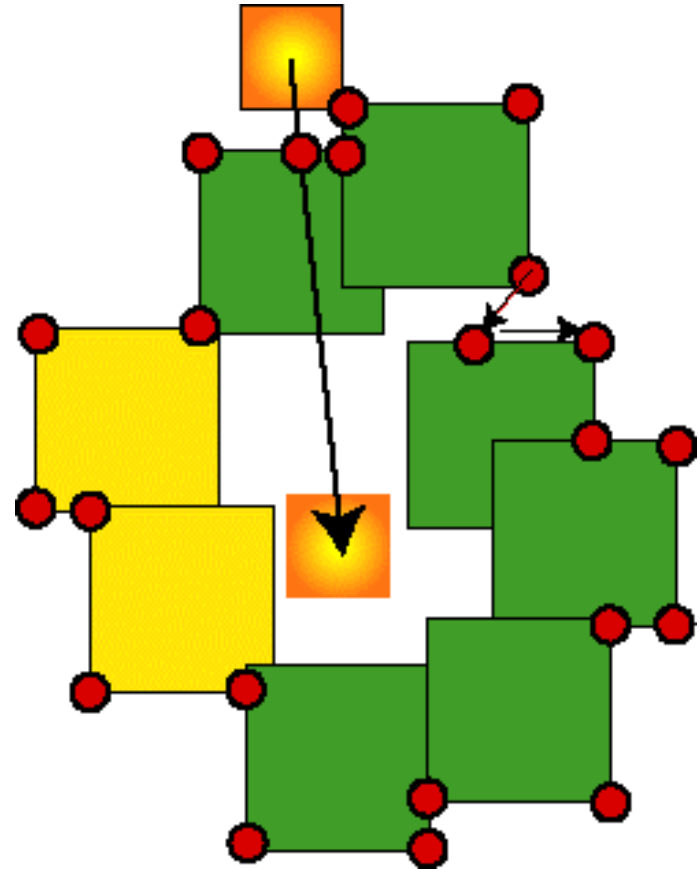
Forbedringer og udvidelser



- Forbedringer
 - Efterfølgende optimering af sti
 - Forgrening med/mod uret
 - 2D-Hashtabel: grid / divide & conquer
 - Tilføjelse af acceleration har kun betydning for hvor hurtigt man kan bevæge sig gennem sving
- Udvidelser
 - Polygon forhindringer
 - Konvekse hylster (hvis du vil et sted hen hvor du ikke lander indenfor det konvekse hylster)
 - Heuristik til at tage højde for objekter der bevæger sig ved at "flytte" lidt på dem og/eller gøre dem længere i bevægelses retningen
 - Heuristik til at vurdere hvor "farlig" en sti er

Cykler igen

3. Cykler der opstår fordi vi altid går med uret, disse cykler kan dog elimineres ved at gå mod uret i disse tilfælde



Stifinder

Så er det bare derudaf!

