Deformerbara kroppar (intro)

Deformerbara objekt utan ben?

- Tyg
- Skumgummi
- Cellplast
- Gelé
- Ballonger
- Lera
- Vatten

Många olika tekniker

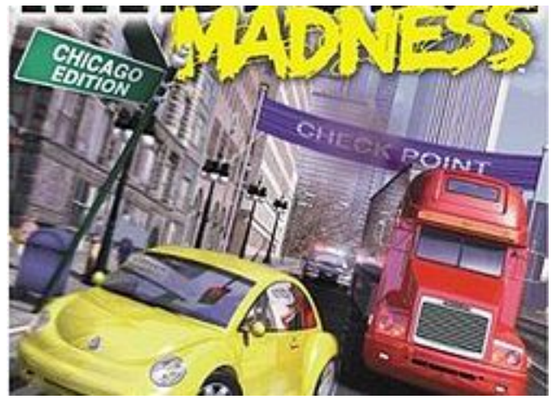
- **Förgenererade deformationer**
- **Massa-fjäder-system**
- **Finita elementmetoden**
- **Formmatchning**
- **Tryckmodellen**
- **Implicit modellering (level sets)**
- **Punktbaserad animation**

Förgenererade deformationer

Vanliga i spel (t.ex Midtown Madness)

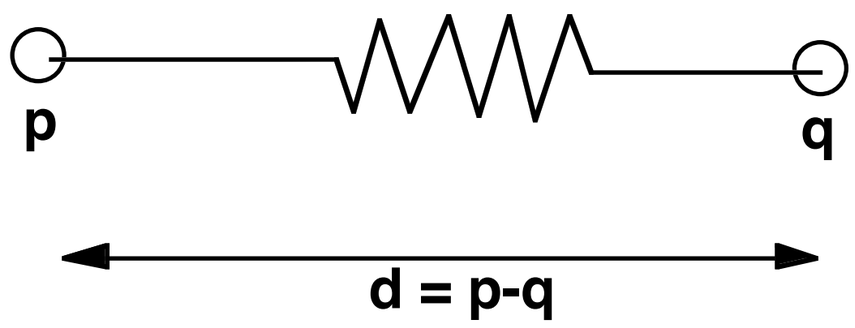
En ren designfråga.

Orealistiskt!



Massa-fjäder-system

En uppsättning punktmassor anslutna med fjädrar.



$|\mathbf{d}| = |\mathbf{p}-\mathbf{q}|$

$\mathbf{f} = -k_s(|\mathbf{d}|-r)$

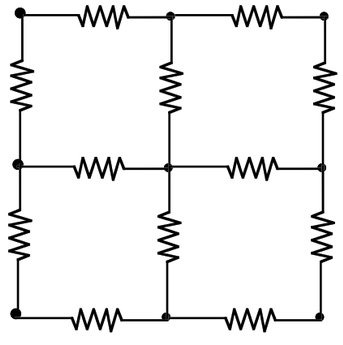
$\mathbf{f} = -k_s(|\mathbf{d}|-r) \cdot \mathbf{d}/|\mathbf{d}|$

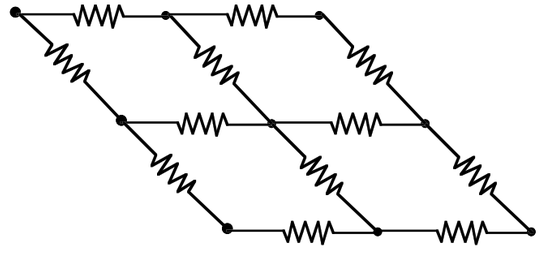
Dämpning kräver även en komponent som är hastighetsberoende.

Massa-fjäder-system

Många anslutningar krävs

Detta duger inte:





BÅDA diagonalerna brukar tas med, samt längre fjädrar ett steg längre ut.

Massa-fjäder-system

Bra i 2D, för tyg mm

Hyfsat för skumgummi etc, men beräkningstungt

Svårt att klara självkollissioner

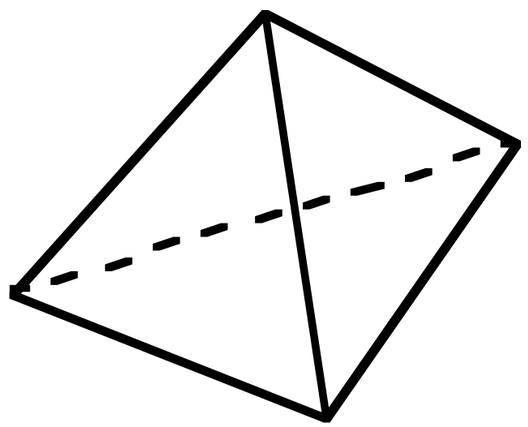
Kan vändas fel utan att det upptäcks

Stabilitetsproblem!

Finita elementmetoden

Bryt ner modellen i tetraedrar

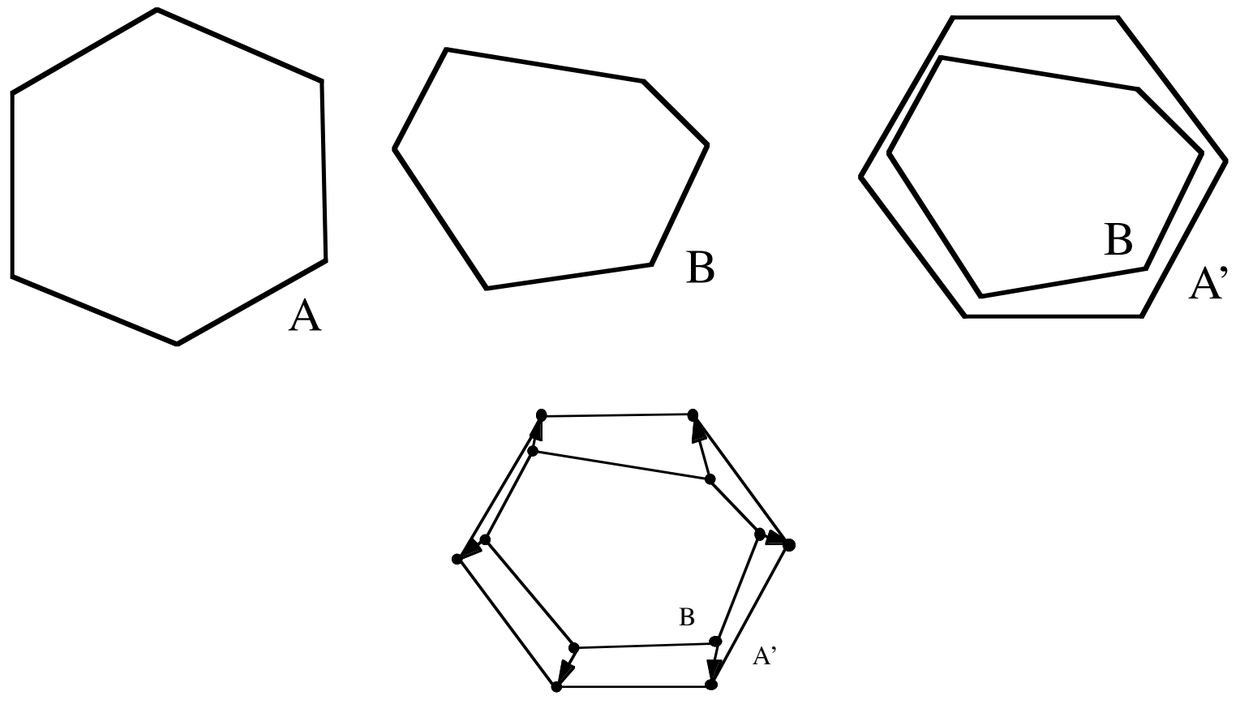
Bra metod, klarar självöverlappningar, inga problem med felaktiga tillstånd.



Formmatchning

Modellen attraheras av ett viloläge

Intressant metod, kan garanteras vara stabil

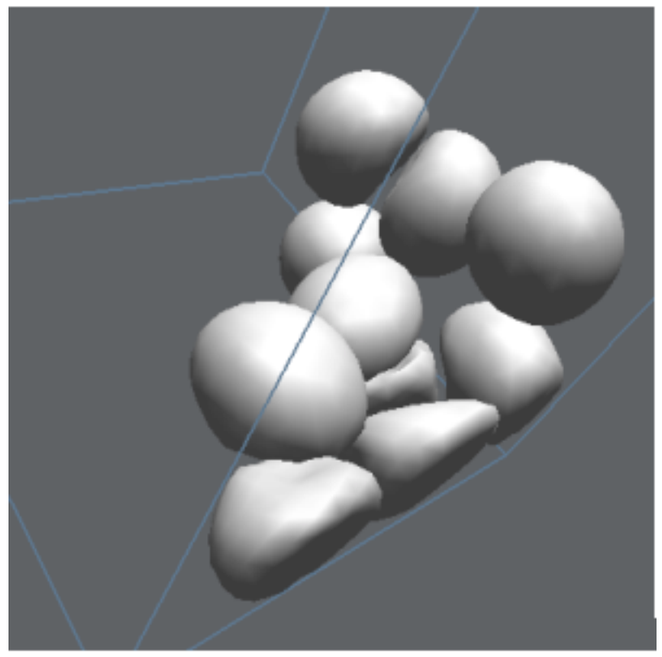


Tryckmodellen

Massa-fjädersystem som 2D-skal plus tryck för innehållet

Mycket bra simulering av

- **Vattenballonger**
- **Celler**



Implicit modellering
Punktbaserad animation

Stora partikelsystem kan simulera vatten, modellera mm mycket realistiskt.

I 2D: iPhone-öl-demot!

Villkorssystem (Constraints)

Animation av deformerbara kroppar med (mer eller mindre) enkla villkor!

Max/min-avstånd
Max/min-vridning

Ragdollanimation

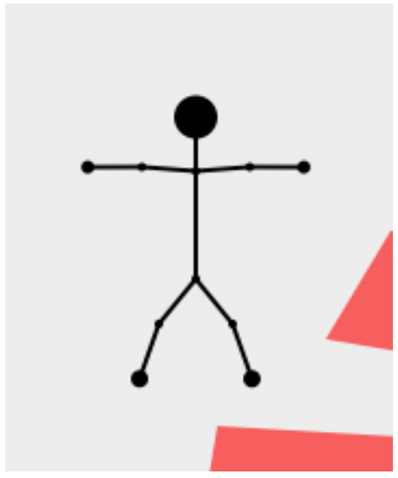
Exempel på

- **Villkorssystem (constraints)**

- **Verletintegrering**

Enkel ragdollmodell

Ett antal punktmassor



Avståndsvillkor

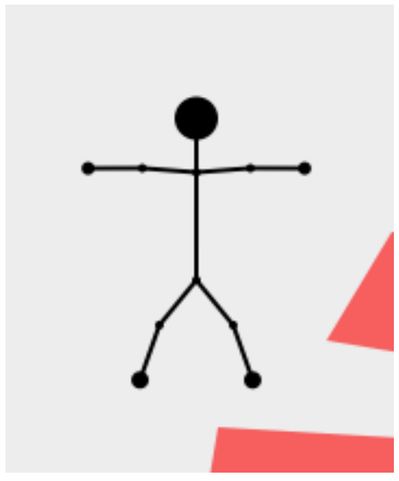
Vridvillkor

Verletintegering

Kollisionsdetektering

Punktmassor

Massa + position + förra position



```
a := NewParticle(100, 25, 1, 10); // Head
b := NewParticle(50, 50, 1, 3); // Hand
c := NewParticle(75, 50, 1, 2); // Elbow
d := NewParticle(100, 50, 1, 2); // Shoulder
e := NewParticle(125, 50, 1, 2); // Elbow
f := NewParticle(150, 50, 1, 3); // Hand
g := NewParticle(100, 100, 1, 2); // Hip
h := NewParticle(90, 125, 1, 2); // Knee
i := NewParticle(110, 125, 1, 2); // Knee
j := NewParticle(80, 150, 1, 4); // Foot
k := NewParticle(120, 150, 1, 4); // Foot
```

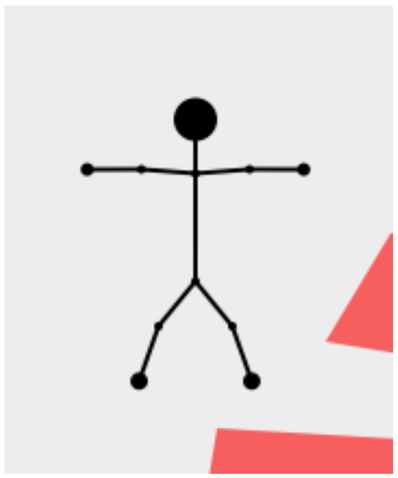
Avståndsvillkor

Parvis, ange ett avstånd mellan två punktmassor

```
NewJoint(a,d); // Head and shoulder
      NewJoint(b,c); // Arms
          NewJoint(c,d);
          NewJoint(d,e);
          NewJoint(e,f);
 NewJoint(d,g); // Shoulder to hip
      NewJoint(g,h); // Legs
          NewJoint(g,i);
          NewJoint(h,j);
          NewJoint(i,k);
```

Animation

Stega partiklarna (Verlet)
Uppdatera avståndsvillkor + hantera kollisioner
Tack vare Verletintegrering krävs inget mer än förflyttning av punktmassor



```
UpdateParticles(dt);
UpdateSticks(dt);
UpdateForces;
```

UpdateParticles med Verletintegrering:

```
p[i].acceleration = p[i].force / p[i].mass;
o = p[i].position;
p[i].position = 2*p[i].position - p[i].oldpos + p[i].acceleration*dt;
p[i].oldpos = o;
```

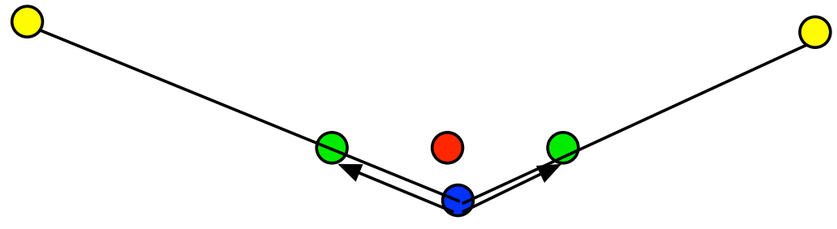
Avståndsvillkor

Parvis, flytta partiklar så deras avstånd blir korrekt.
Bör göras ”parallellt” då sekvensiell uppdatering inför assymetrier

För två kopplade punkter i och j

Beräkna nya kandidatpunkter genom att flytta båda till rätt avstånd

Varje punkts nya position är medelvärdet av alla kandidater

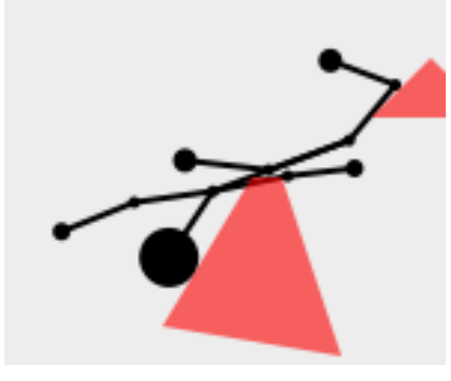


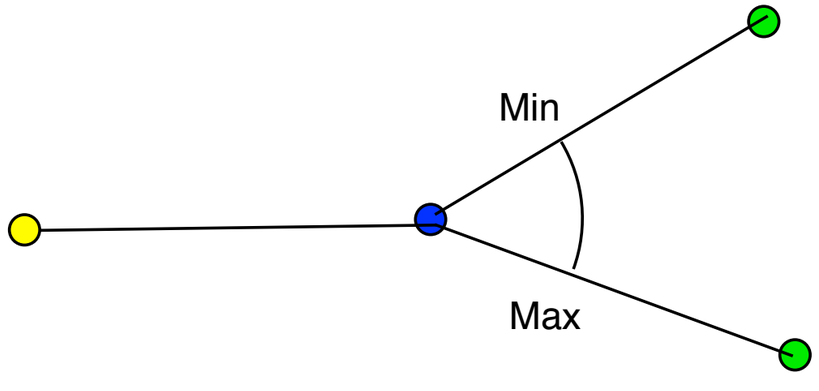
Vridvillkor

Vi har ny en hyfsad slapp ragdoll/trasdocka. Brist på vridvillkor gör vår ”docka” helt slapp och ostyrig.

Alla leder är kulleder, dockan fäller ihop sig som en hög spaghetti.

Inför vridvillkor: Serie av tre punkter får ett tillåtet vridområde. (Kan i 2D vara två vinklar kring π.)

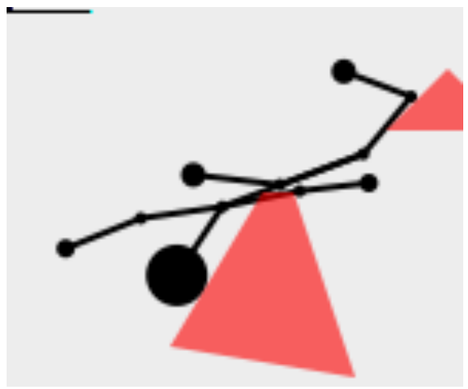


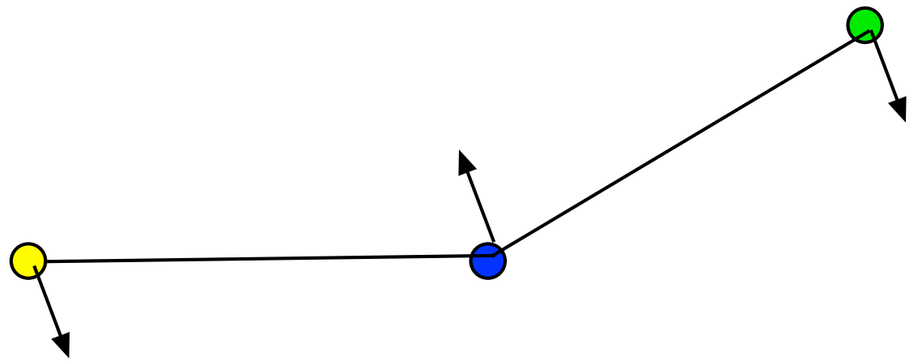


Uppfyllande av vridvillkor

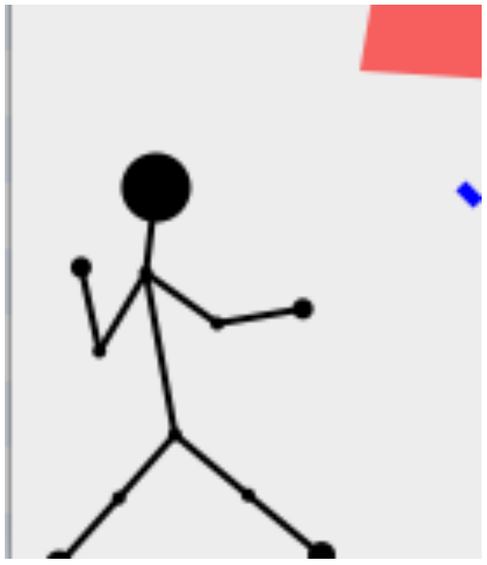
Vridvillkor kan inte uppfyllas genom att flytta *en* punkt.

Samtliga punkter flyttas så rörelsemängden bevaras.





Resultat med vridvillkor



Tämligen pigg och stabil modell.

Kan nästan stå för sig själv.

3D givetvis svårare.

Slutsater:

• Verletintegrering gör många fysiksimuleringar enklare, lättare att få stabila.

• Villkorssystem vitala för många fall.

Deformerbara kroppas så här långt

Animation med fysik:

- Massa-fjäder-system
- Andra metoder som formmatchning och tryckmodellen
- Villkorssystem

Nästa gång

3D-system

Förstörbara objekt

Vätskor och gaser

Deferred shading