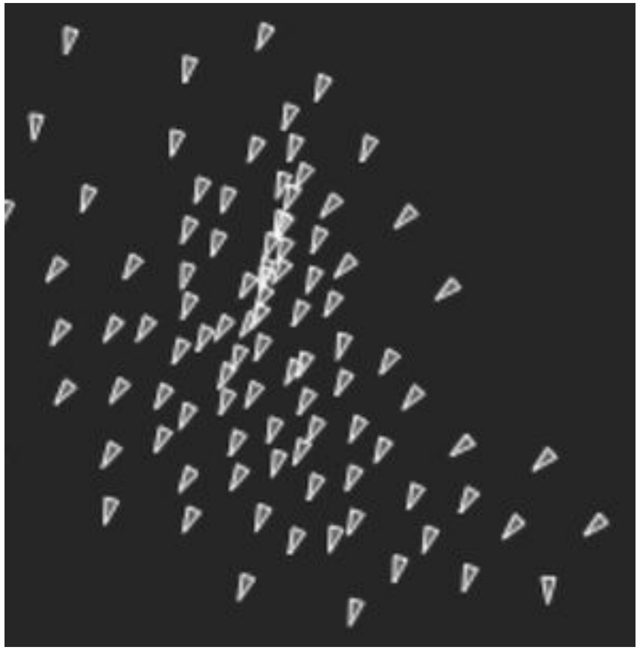
CNAM - NSY116 - déc. 2013

Animation interactive

Pierre Cubaud <cubaud @ cnam.fr>



- Généralités
- Easing
- Dynamique du point
- Particules, nuées
- Masse-Ressort
- Structures articulées

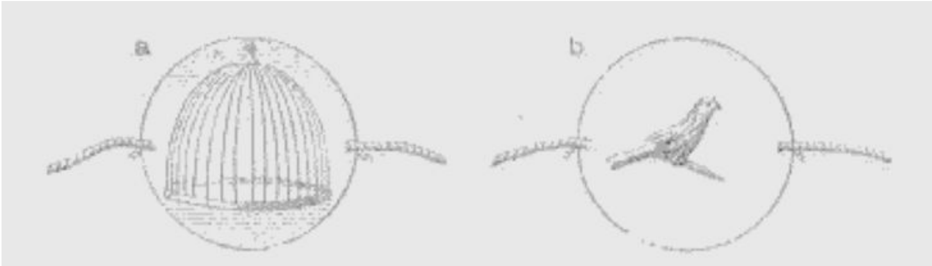
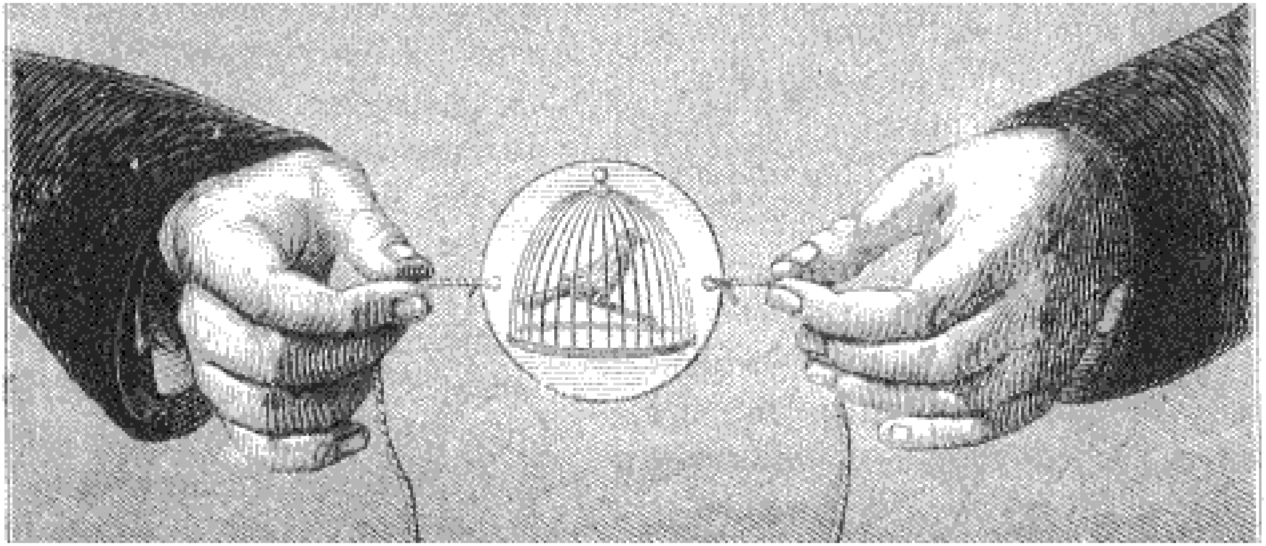


Fig. 1. Disque du thaumatrope vu en dessus (*a*) et en dessous (*b*).



revue *La Nature* http://cnum.cnam.fr

1829 - Joseph PLATEAU (Belgique) - Thaumatrope

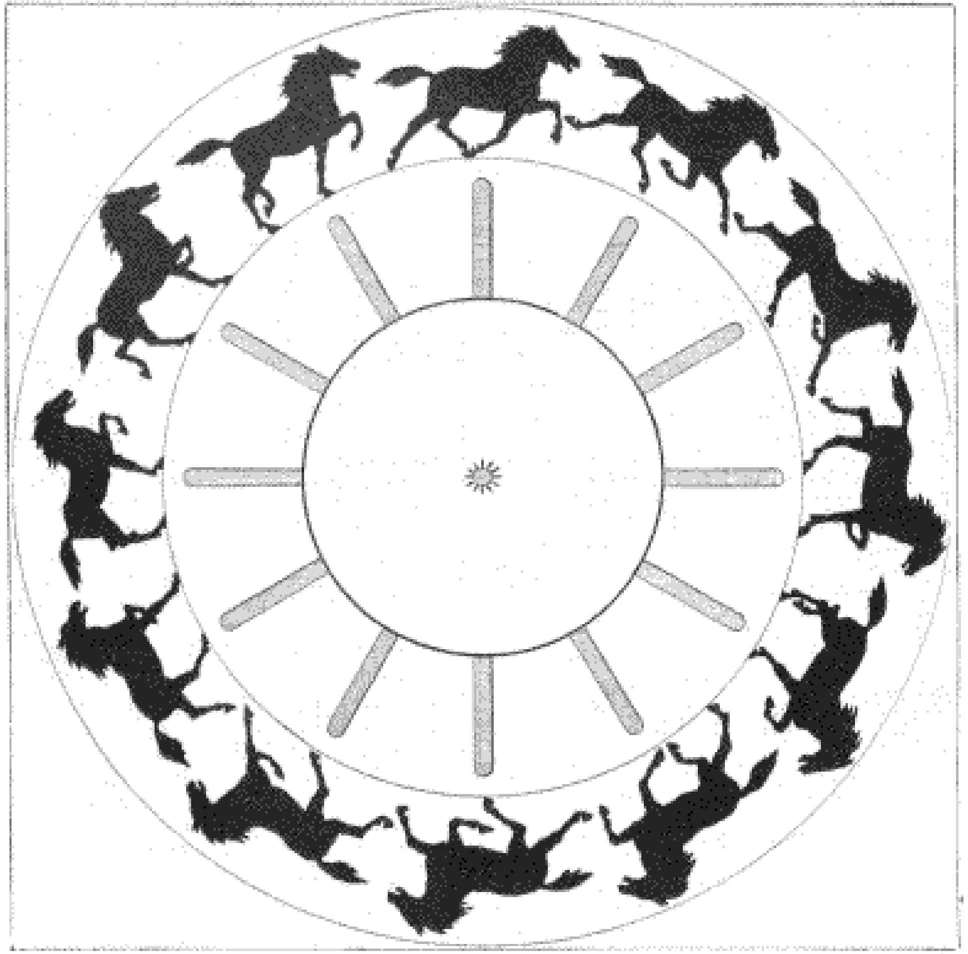
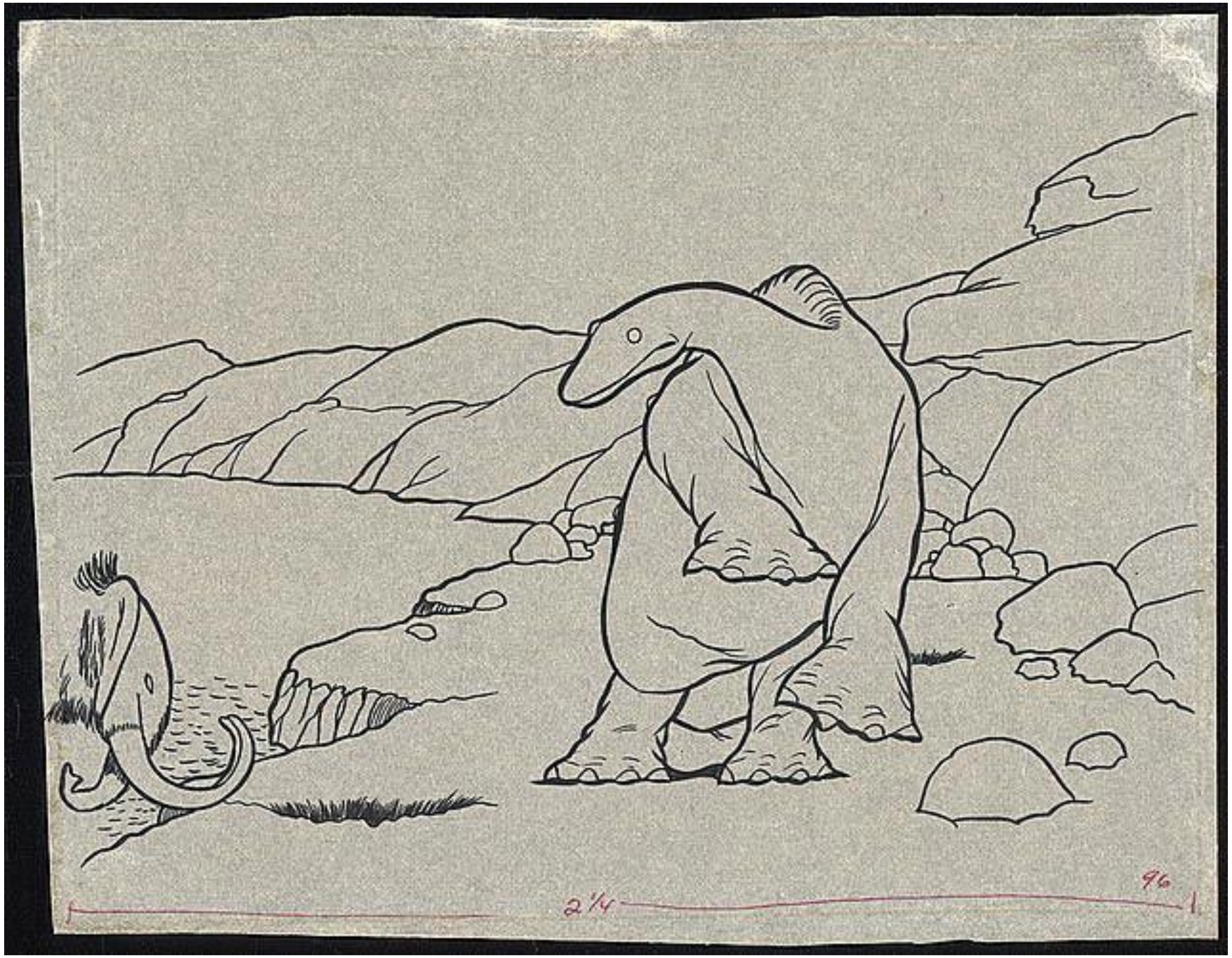


Fig. 1. Disque zootropique d'un cheval au trot. (D'après les photographies instantanées de M. Muybridge.)

revue *La Nature* http://cnum.cnam.fr

E. MUYBRIDGE - Zoopraxinoscope

1895 - Auguste et Louis LUMIERE (Lyon)- Le cinématographe



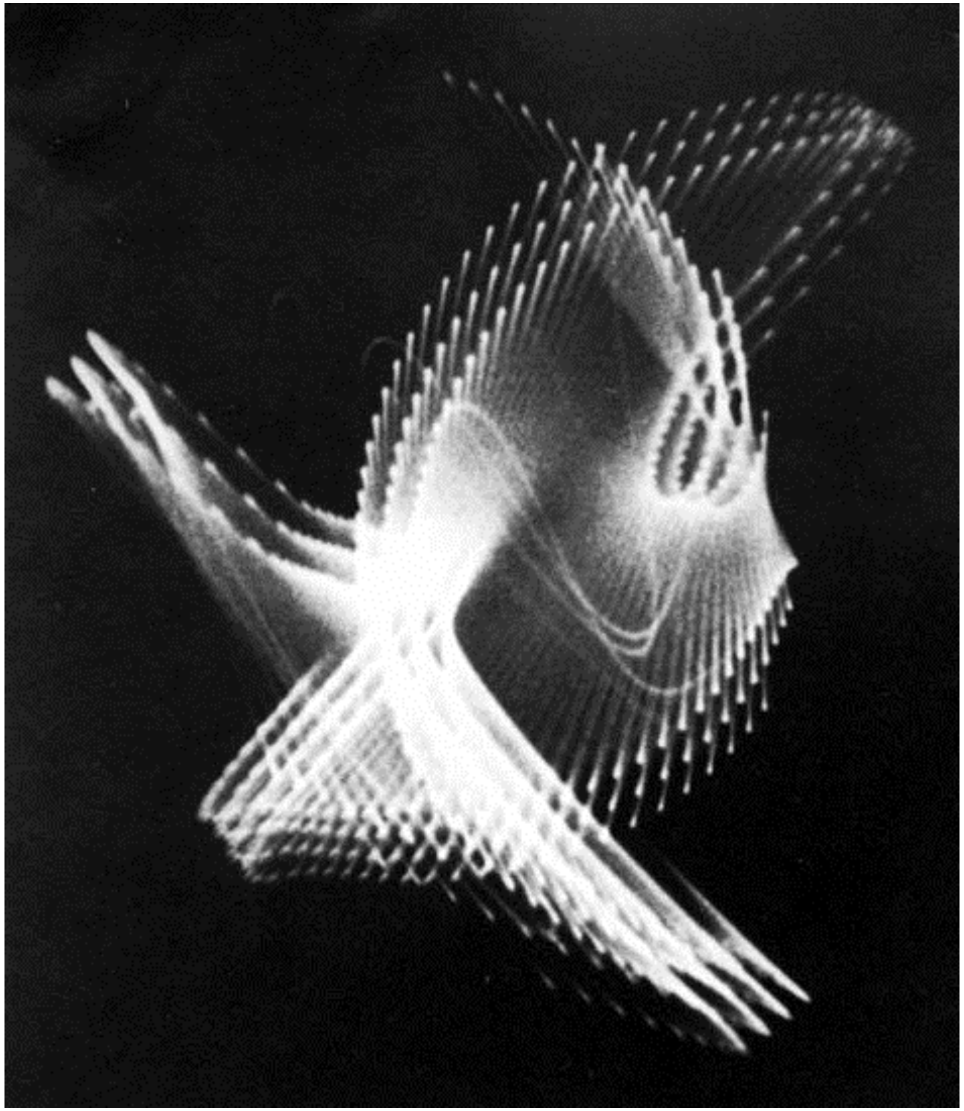
http://www.loc.gov/rr/print/swann/artwood/aw-animation.html

ca 1910 - Windsor McCAY - Gertie the dinosaur

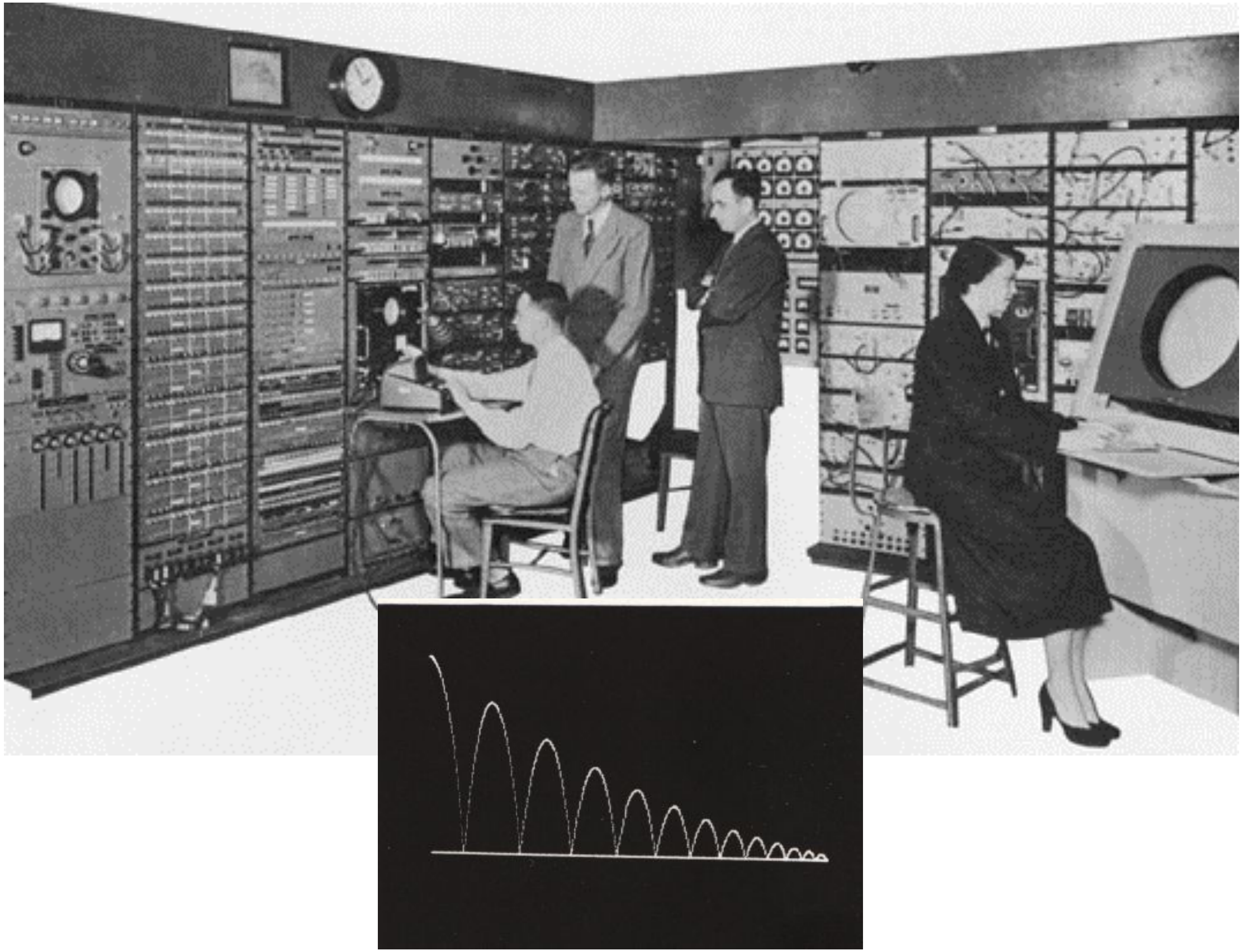
1937 - Walt DISNEY - Snow White

Multiplane camera

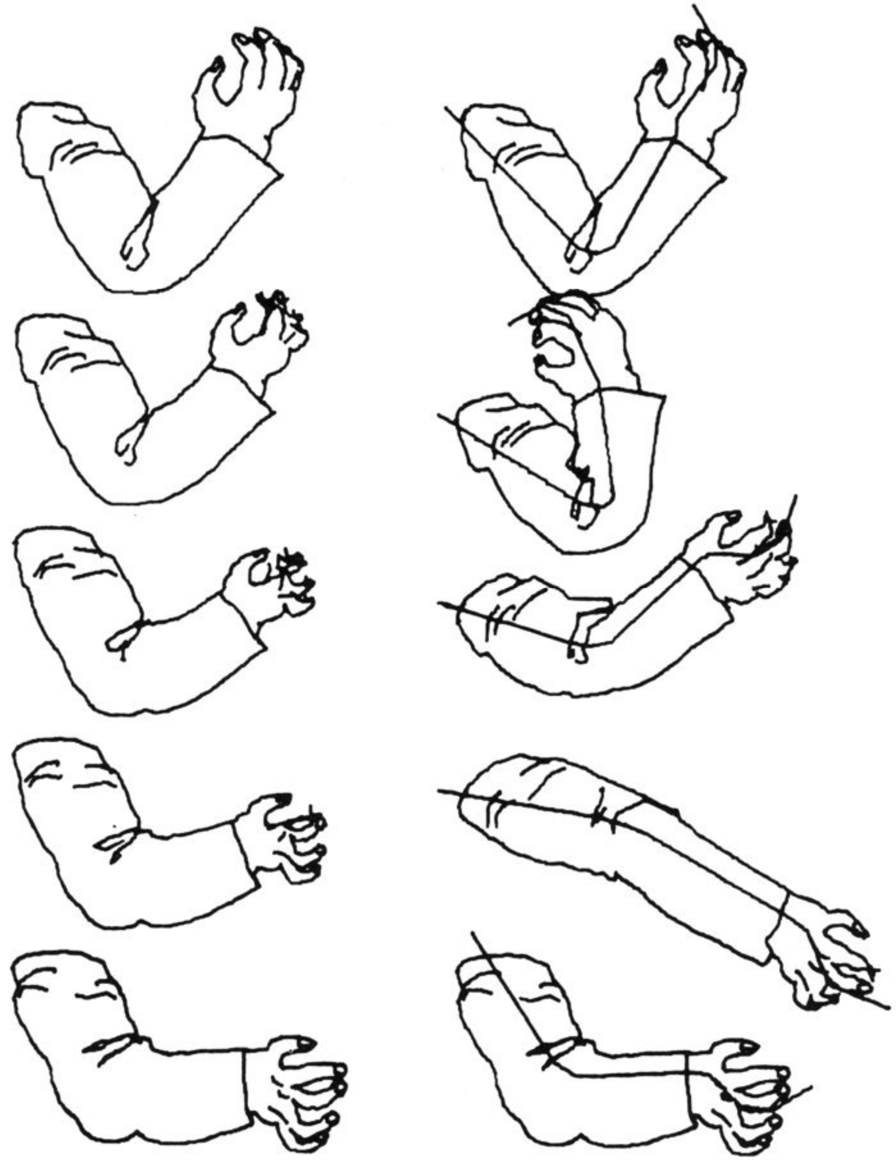
(musée du Walt Disney studio, Burbank)



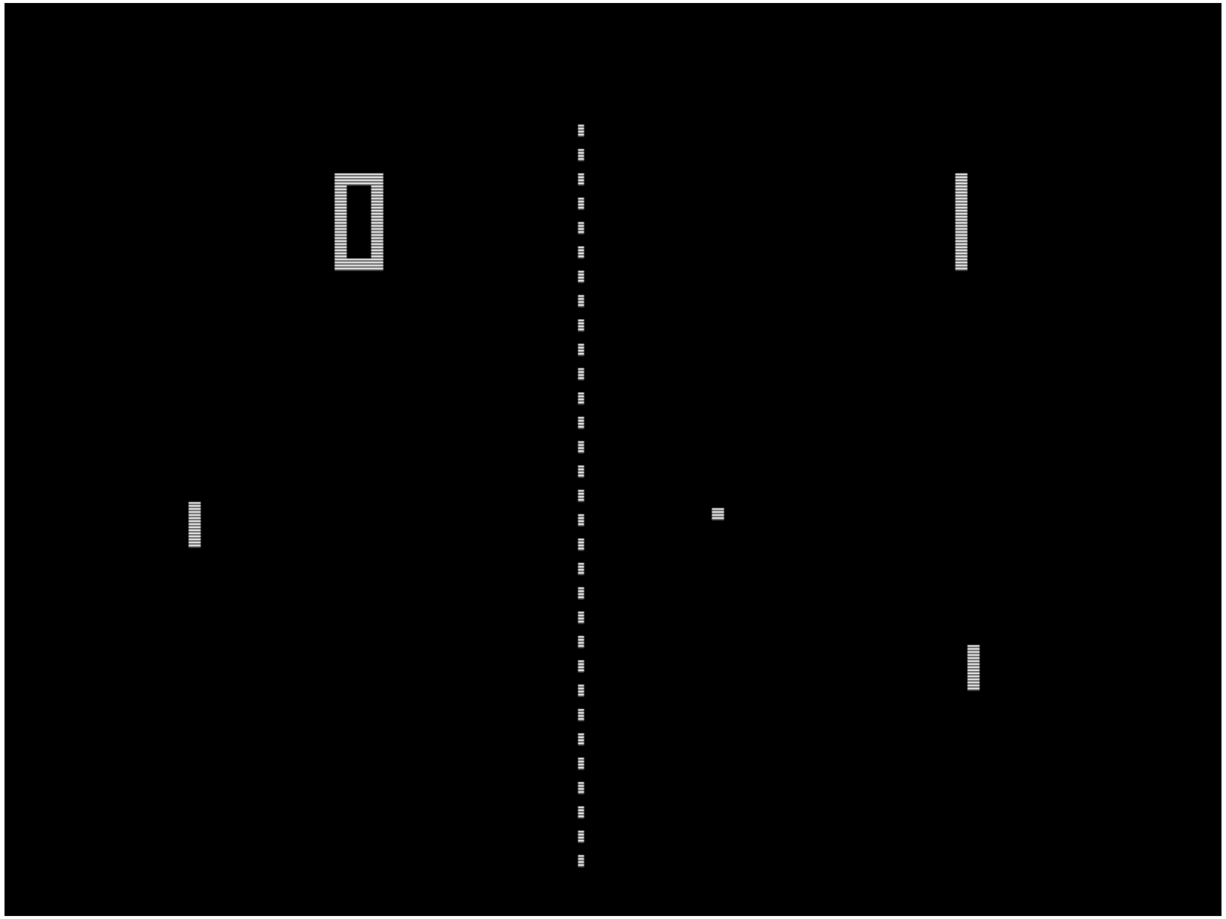
1950 - Ben Laposky



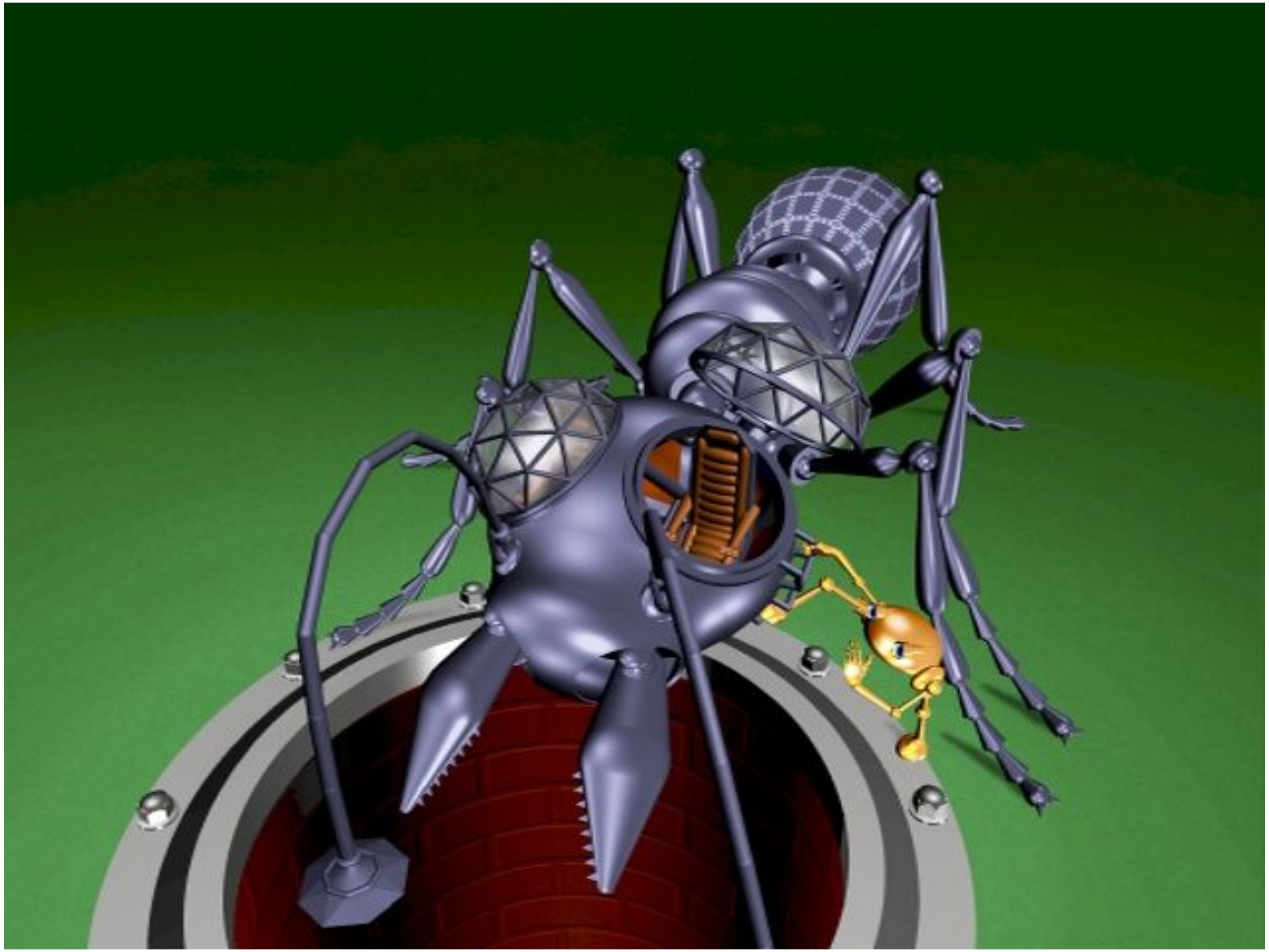
1949-51 - Projet WHIRLWIND



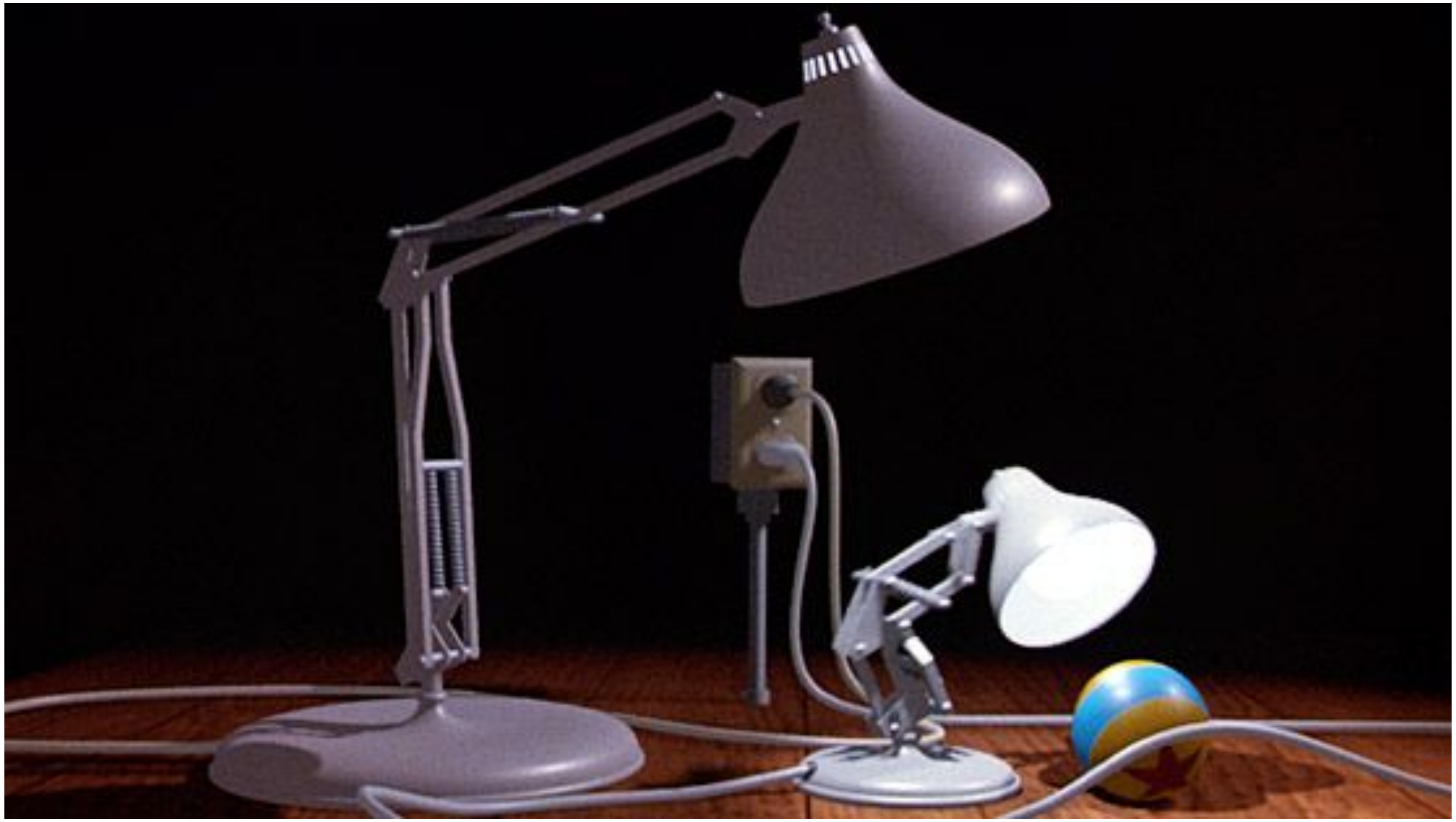
1971 - BURTNYK & WEIN (Canada)



1972 - Pong (Atari)



1980 - NYIT - The works

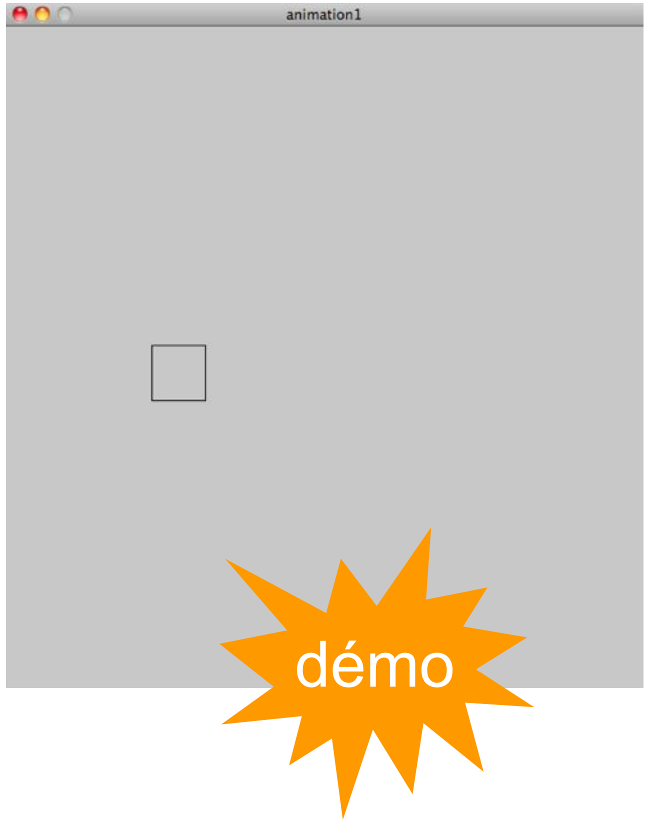


1986 - Pixar (Lasseter et al.) - Luxo junior

Une première animation (ratée)

demo9modereactifBIS

```
void setup() {
  size(600,600);
  noFill();stroke(0);
}

void draw() {
  background(200);
  float x=random(0,width);
  float y=random(0,height);
  rect(x,y,50,50);
}
```



pourquoi voit-on en fait plusieurs carrés ??

le mouvement brownien

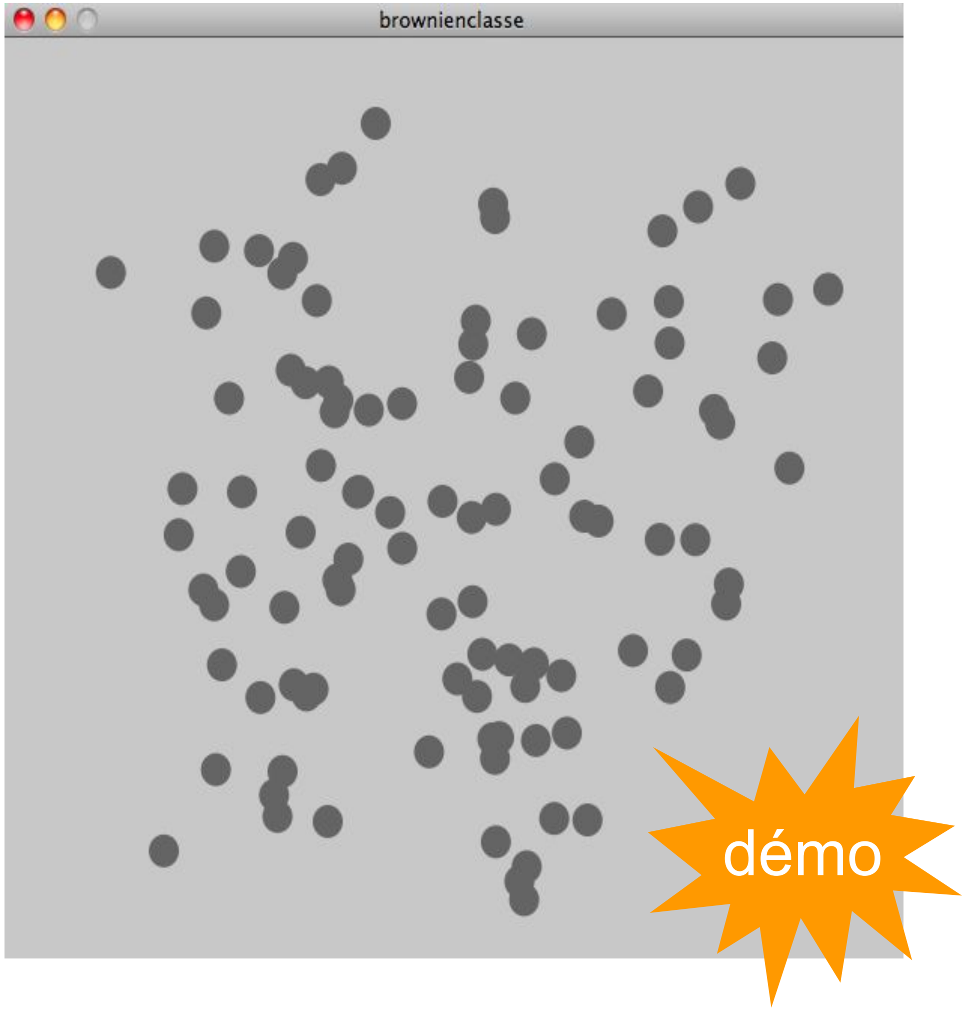
animation2

```
float x=300;
float y=300;

void setup() {
  size(600,600);
  fill(100);noStroke();
  smooth();
}

void draw() {
  background(200);
  x = x + random(-2,+2);
  y = y + random(-2,+2);
  ellipse(x,y,50,50);
}
```

démo

changement de position
par petit incrément aléatoire

plusieurs particules ? passer à l'objet



initialisation et boucle d'animation :

brownienclasse

```
particule[] p = new particule[100];

void setup() {
 size(600,600);
 frameRate(15);
 smooth();
 for (int i=0;i<p.length;i++) p[i] = new particule();
}

void draw() {
 background(0);
 for (int i=0;i<p.length;i++){
  p[i].dessin();
  p[i].evolution();
 }
}
```

```
class particule {
  float x,y;

  particule(){
    x=300; y=300;
  }

  void dessin(){
    fill(100);noStroke();
    ellipse(x,y,20,20);
  }

  void evolution(){
    x = x+random(-10,+10);
    y = y+random(-10, +10);
  }
}
```

- tester des variantes selon la position initiale, les options de dessin
- étudier l'influence du frameRate()

L'animation dans l'IHM

eviter les changements brusques : une ref ?

ex sur tactile : coverflow? annonce la physique

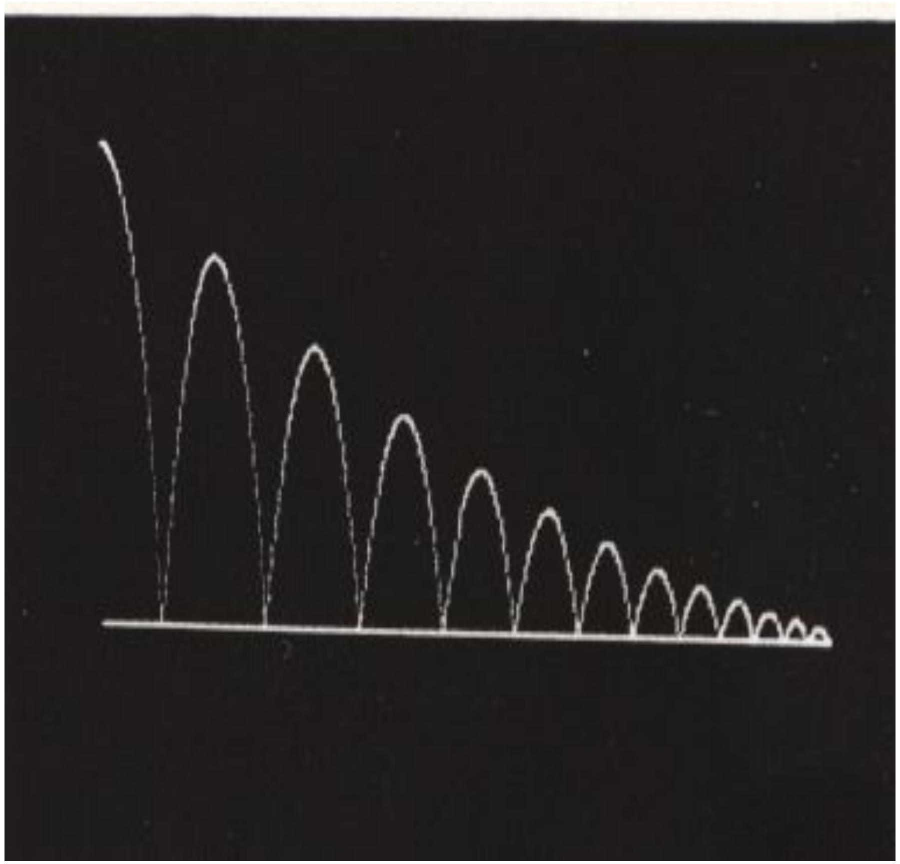
Le "easing"

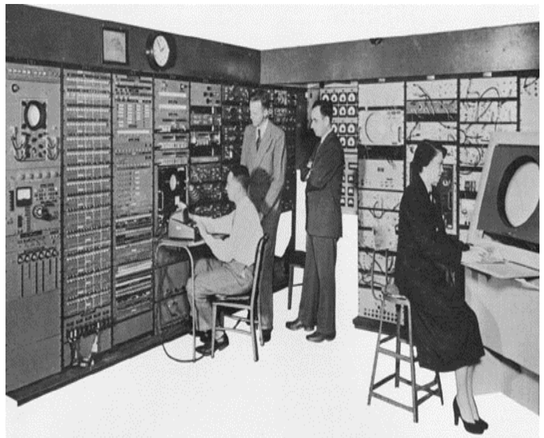
easingcurseur

```
// d'apres Geridan & Lafargue "Processing"
float x=0;
float y=0;
float ease=0.1;

void setup(){
  size(400,400);
  smooth();
  background(0);
}

void draw(){
  background(0);
  x += (mouseX -x)*ease;
  y += (mouseY -y)*ease;
  ellipse(x,y,20,20);
}
```

plusieurs autres méthodes (fonction sinus etc)

Dynamique du point : ex. du boulet de canon (dans le vide)

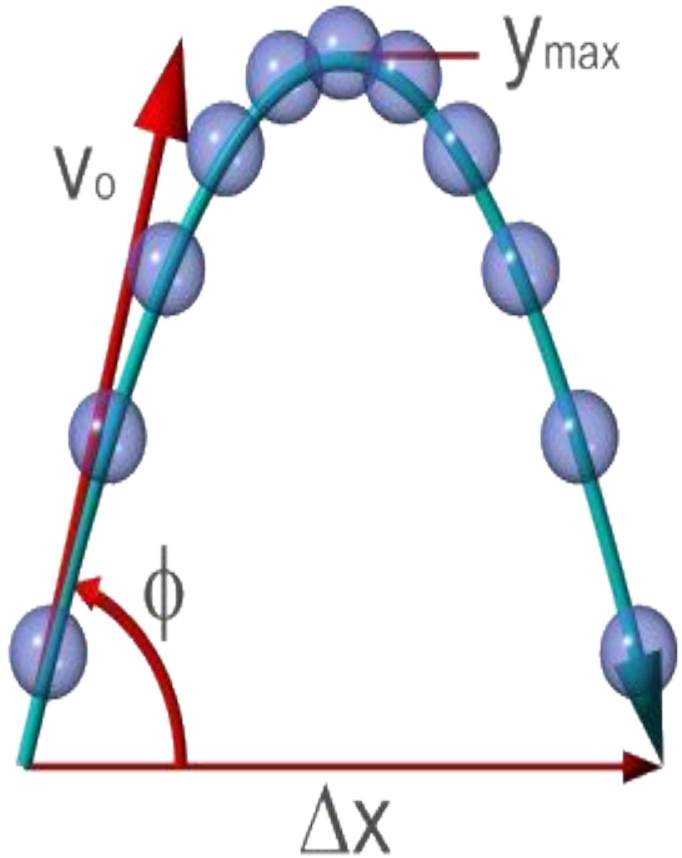




1949-51 - Projet WHIRLWIND

Un peu de physique :

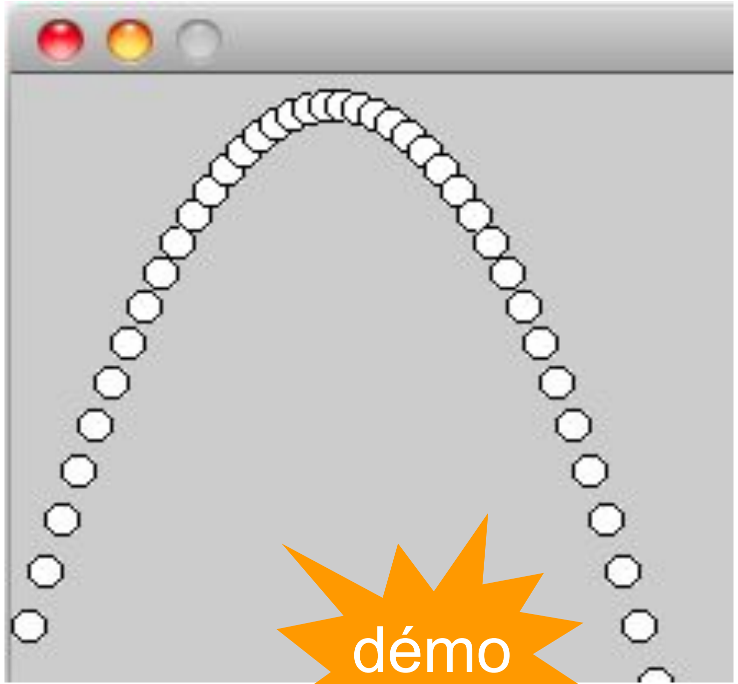
$$\sum_i \vec{F}_i = m\vec{a}$$



$$\sum \vec{F} = \begin{pmatrix} 0 \\ -mg \\ 0 \end{pmatrix} = m\vec{a} = \begin{pmatrix} m\ddot{x} \\ m\ddot{y} \\ m\ddot{z} \end{pmatrix} \Rightarrow \begin{cases} \dot{x} = v_0 \cos\Phi \\ \dot{y} = v_0 \sin\Phi - gt \\ \dot{z} = 0 \end{cases} \Rightarrow \begin{cases} x = v_0 t \cos\Phi \\ y = v_0 t \sin\Phi - \frac{1}{2} g t^2 \\ z = 0 \end{cases}$$

boulet

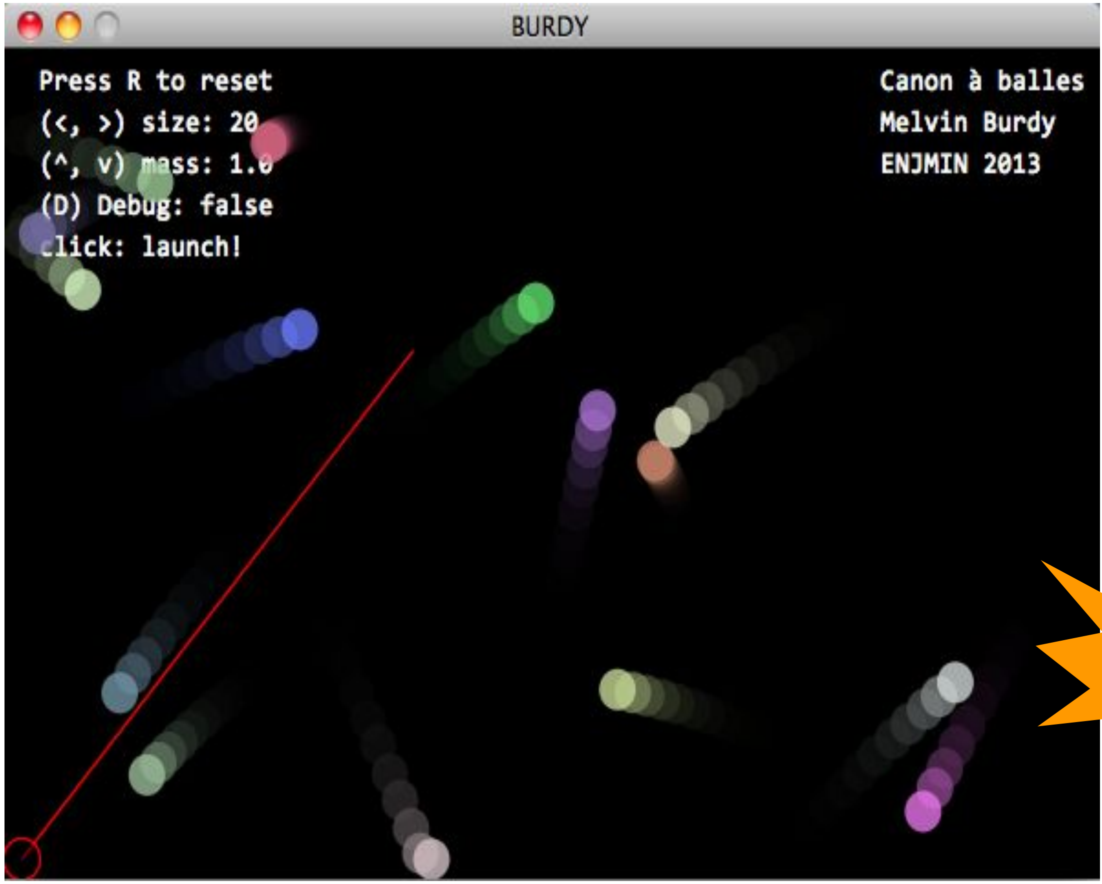
```
float x,y,vx,vy,ax,ay;

void setup(){
  size(600,200);
  x = 0;
  y = 0;
  vx = 5;
  vy = 20;
}

void draw()  {
  ax = 0;
  ay = -1                = 1 m/s2
  vx = vx + ax;
  vy = vy + ay;
  x = x + vx;
  y = y + vy;
  ellipse(x,height-y,10,10);
}
```



démo

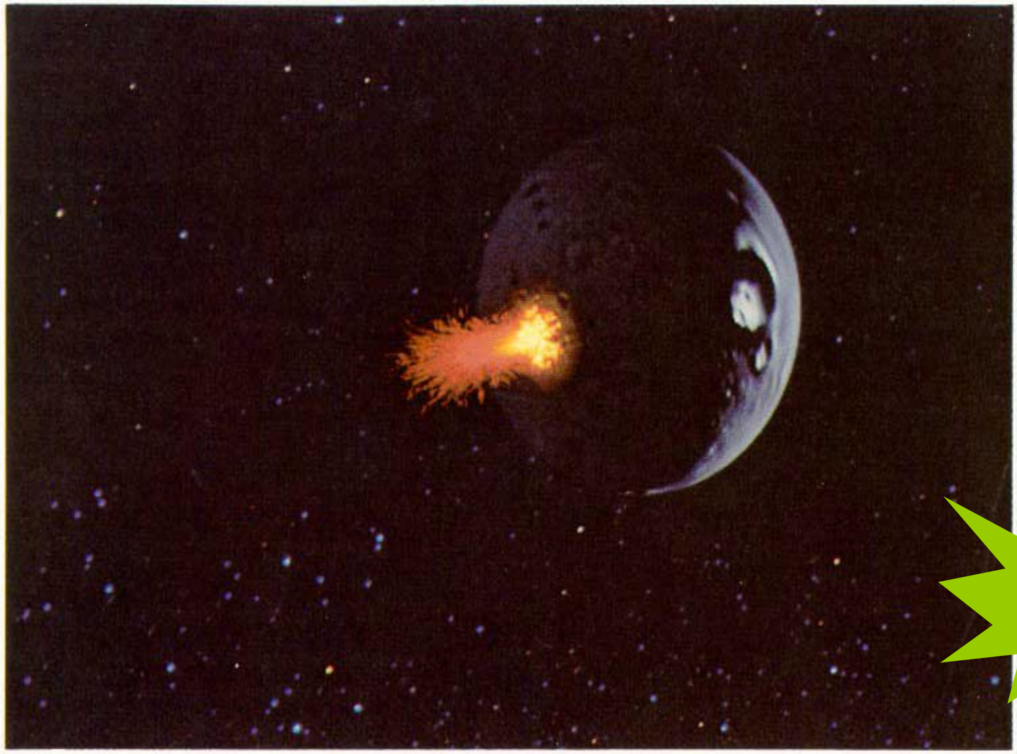
ajouter les rebonds

Prise en compte des collisions entre objets

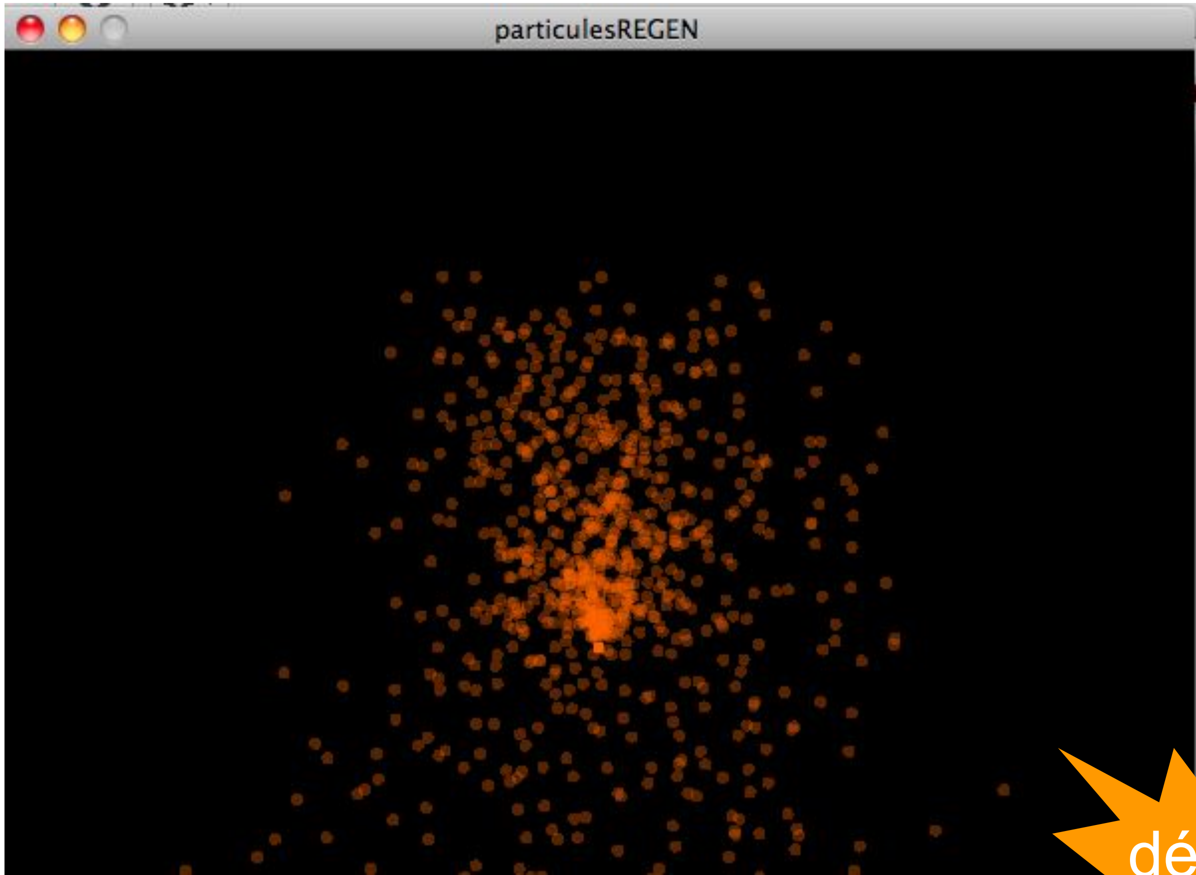


démo

Une extension intéressantes : les particules



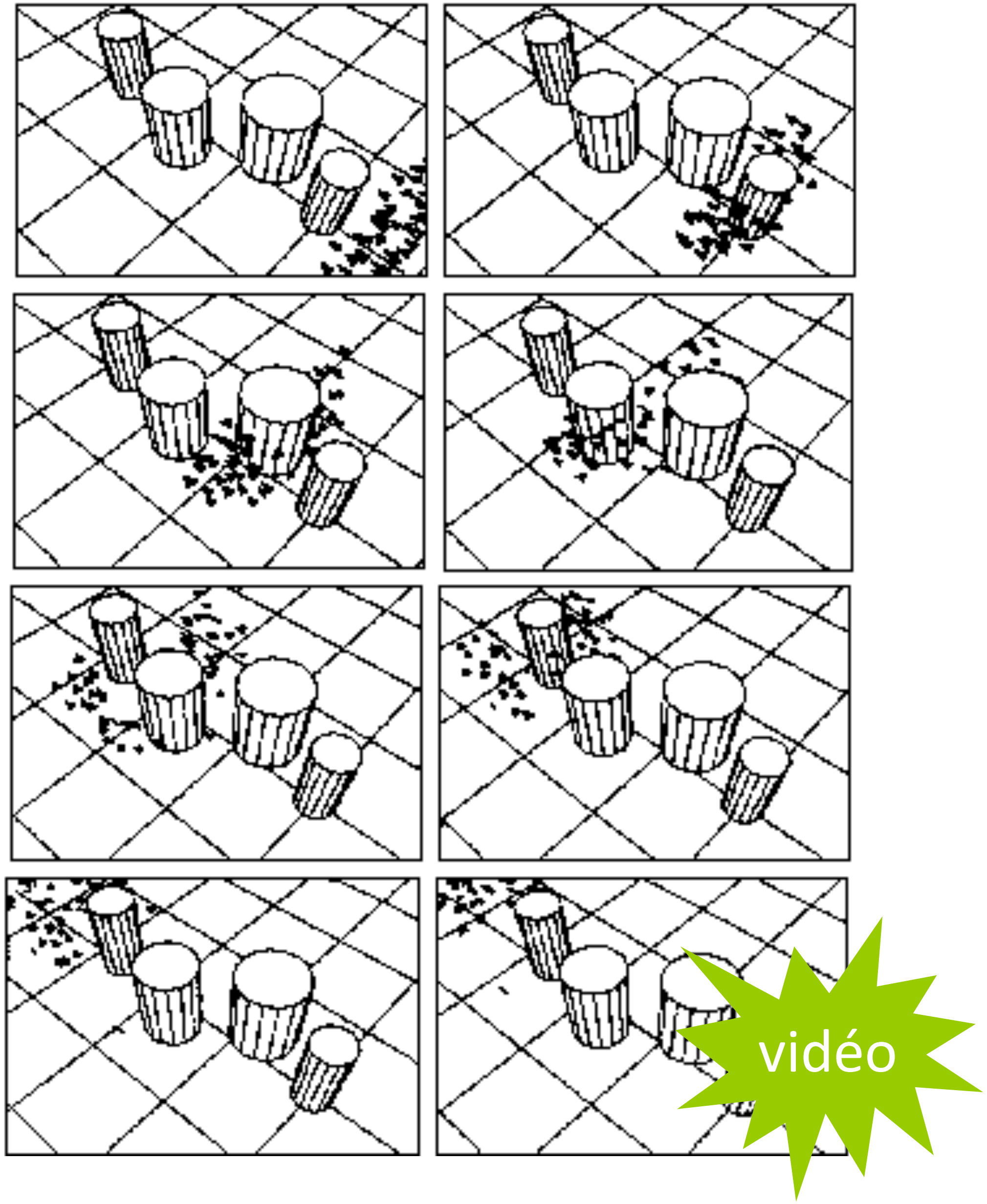
W. T. Reeves, "Particle Systems A Technique for
Modeling a Class of Fuzzy Objects",
Computer Graphics, vol. 17, no. 3, pp 359-376, 1983



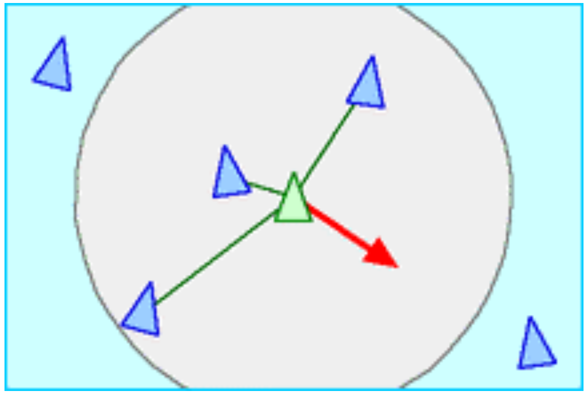
démo

Les nuées (boids)

C. W. Reynolds,
"Flocks, Herds, and Schools:
A Distributed Behavioral Model",
Computer Graphics,
vol. 21, no. 4, pp 25-34, 1987.

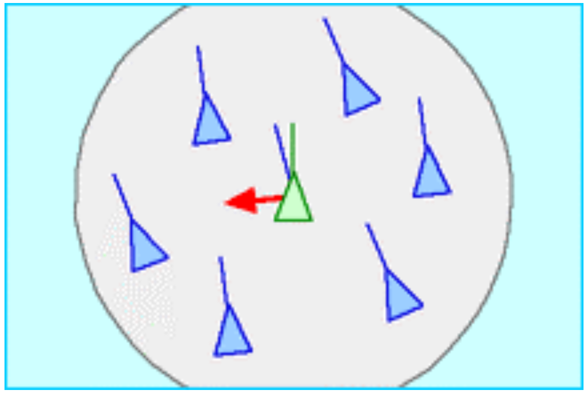


VOIR SON SITE !!
http://www.red3d.com
/cwr/boids

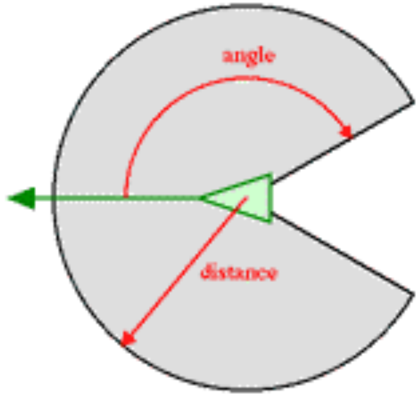


Separation: steer to avoid crowding local flockmates

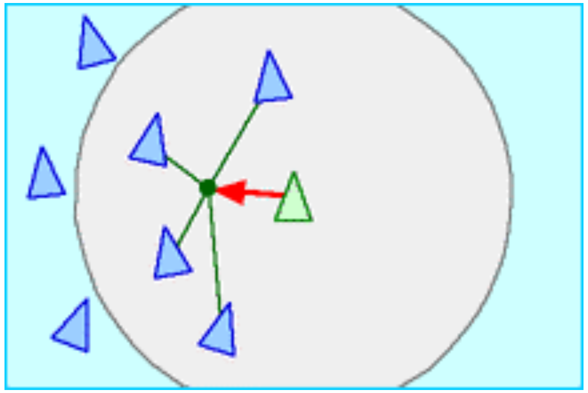
deux forces concurrentes :
-la cohésion au sein du groupe
-l'évitement de collision (prioritaire)



Alignment: steer towards the average heading of local flockmates



a boid's neighborhood



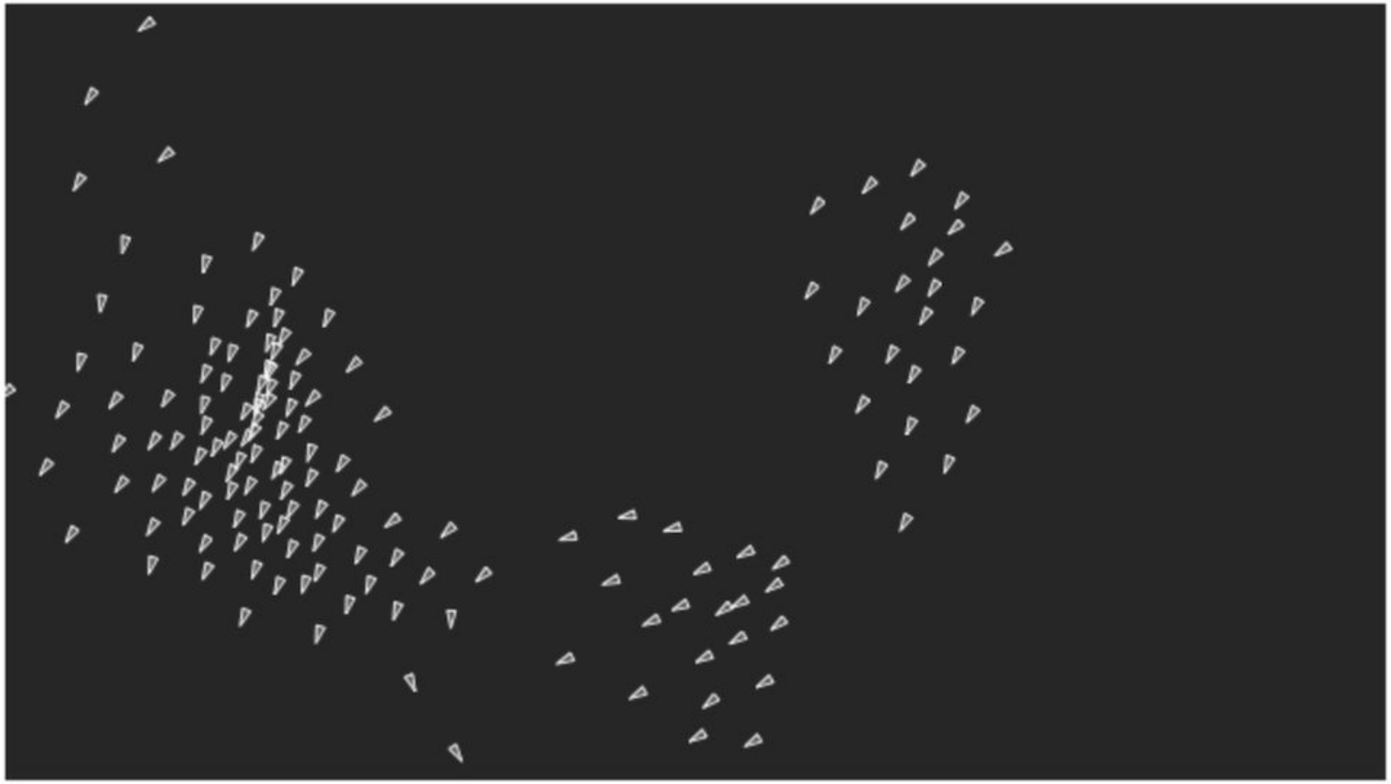
Cohesion: steer to move toward the average position of local flockmates

http://www.red3d.com/cwr/boids

Exemple de programmation avec Processing :

Flocking

This example is for Processing version 1.0+. If you have a previous version, use the examples included with your software. *If you see any errors or have comments, please* *let us know*.

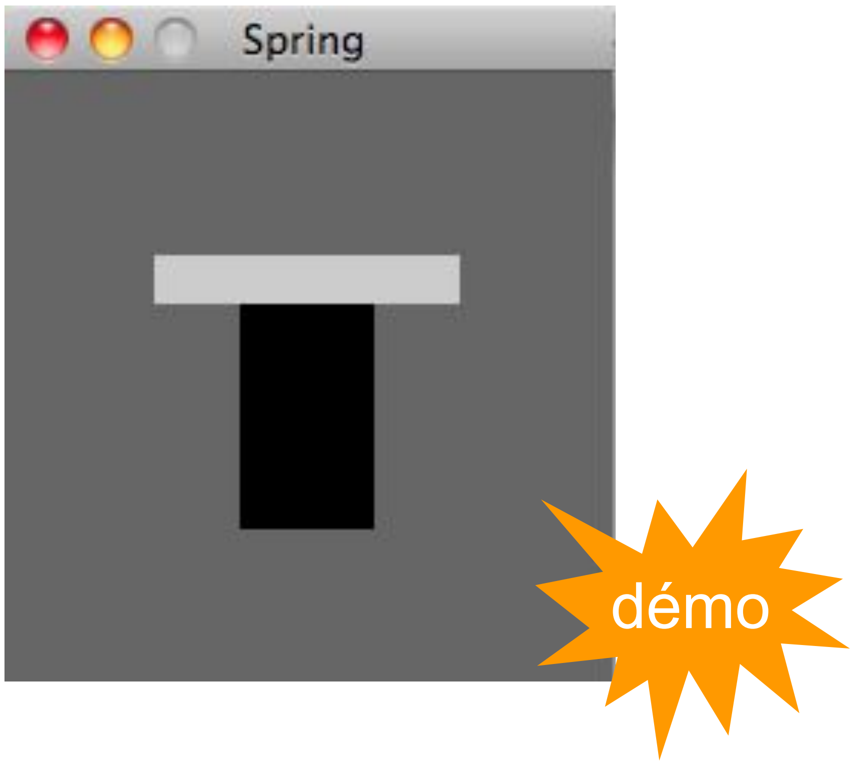


Flocking by Daniel Shiffman.

An implementation of Craig Reynold's Boids program to simulate the flocking behavior of birds. Each boid steers itself

Système masse-ressort

code "Spring" des exemples livrés avec Processing
(rubrique Topics -> Simulate)



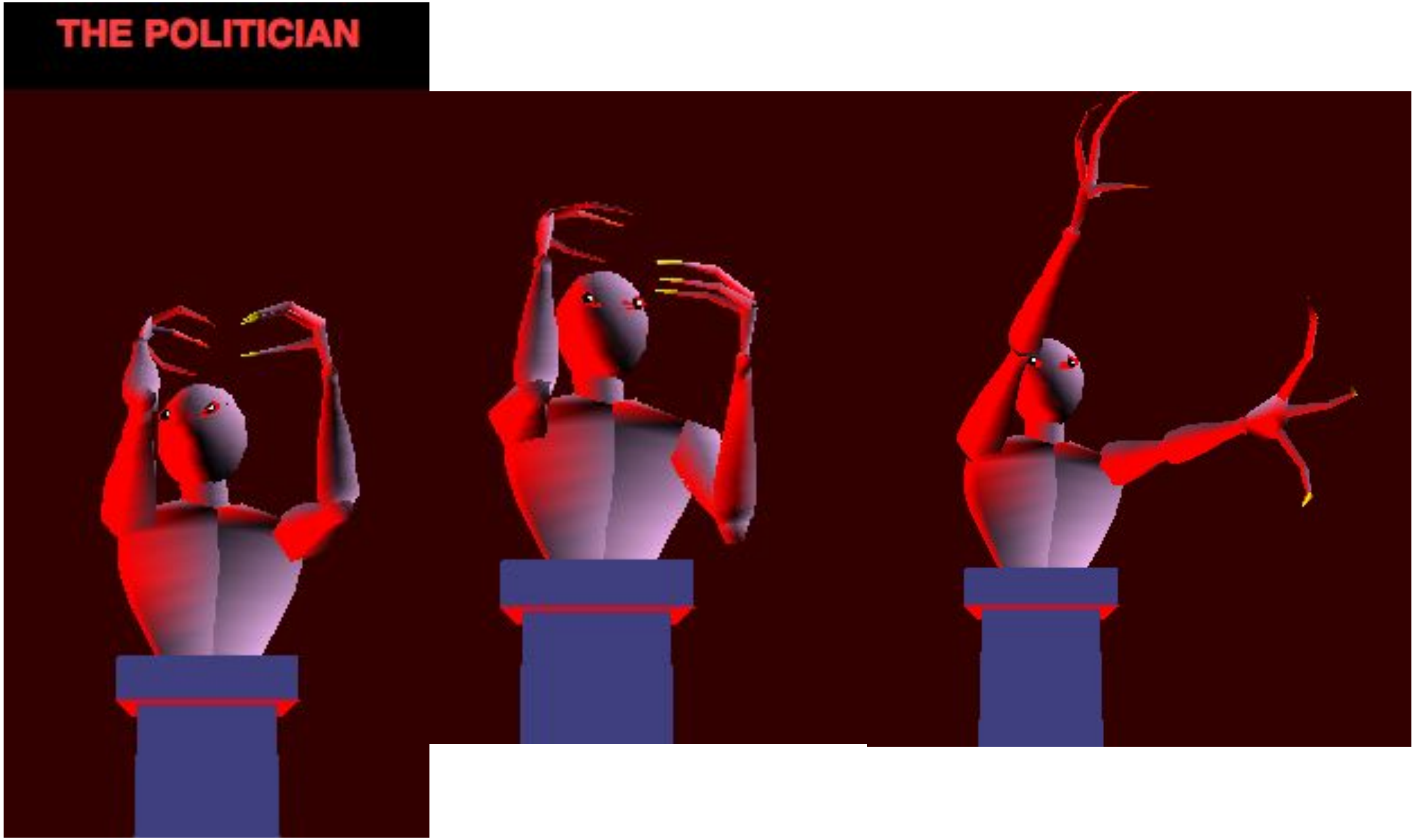
"ut tensio sic vis" : telle extension, telle force (Hooke, 1678)

Spring

```
// Update the spring position
if(!move) {
  f = -K * (ps - R);     // f=-ky
  as = f / M;            // Set the acceleration, f=ma == a=f/m
  vs = D * (vs + as);    // Set the velocity
  ps = ps + vs;          // Updated position
}
if(abs(vs) < 0.1) {
  vs = 0.0;
}
```

- "jouer" avec les paramètres physiques K, m
- rôle du facteur d'amortissement D ?

Structures articulées



http://mrl.nyu.edu/~perlin/experiments/fiend/

Exemple d'un bras manipulateur plan :

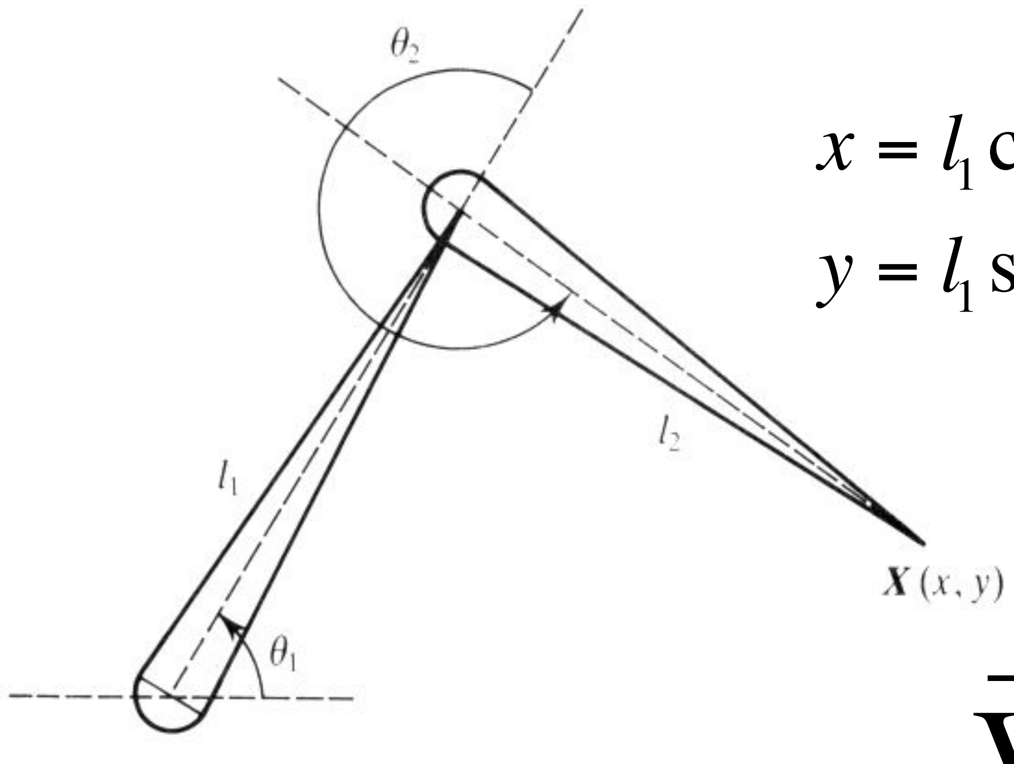


Figure 16.2 A simple two-link structure.

$$x = l_1 \cos\theta_1 + l_2 \cos(\theta_1 + \theta_2)$$
$$y = l_1 \sin\theta_1 + l_2 \sin(\theta_1 + \theta_2)$$

$$\vec{X} = f(\vec{\theta})$$

Cinématique inverse

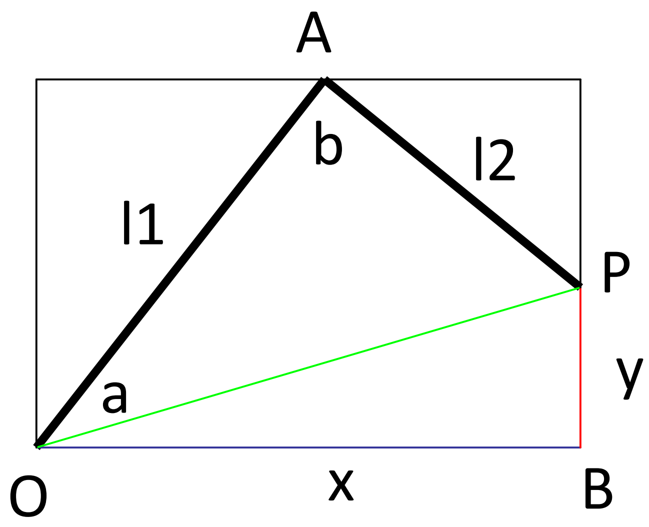
$$\vec{\theta} = f^{-1}(\overrightarrow{X})$$

Reprise de l’exemple :

-recherche angle <BOP>= qT

-recherche angle <POA>= q1-qT=a

-recherche angle <OAP>=p-q2=b



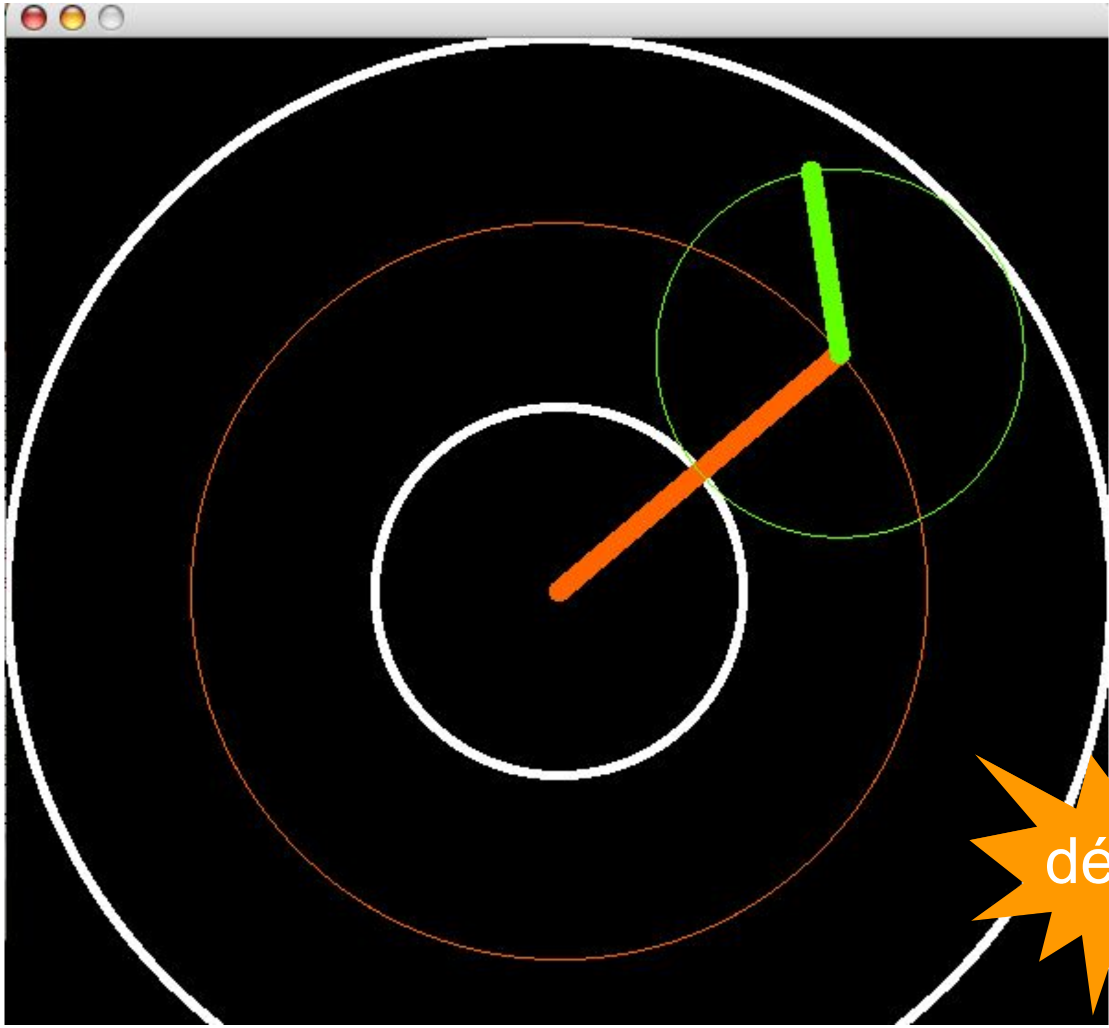
$$tg\theta_T = BP/OB = y/x \Rightarrow \theta_T$$

$$\cos a = \frac{OP^2 + OA^2 - AP^2}{2|OP||OA|} = \frac{d^2 + l_1^2 - l_2^2}{2dl_1} \Rightarrow \theta_1$$

$$\cos b = \frac{AP^2 + AO^2 - OP^2}{2|AP||AO|} = \frac{l_2^2 + l_1^2 - d^2}{2l_1 l_2} \Rightarrow \theta_2$$

avec $d^2 = x^2 + y^2$

Réalisation avec Processing : cineminverse



```
float l1= 200;
float l2= 100;
float x,y,tht,th1,th2,a,b,d;

void setup(){
  size(600,600);
  ellipseMode(CENTER);
  noFill();
}

void draw(){
 background(0,0,0);
 stroke(255,255,255);
 strokeWeight(4);
 ellipse(300,300,2*(l1-l2),2*(l1-l2));
 ellipse(300,300,2*(l1+l2),2*(l1+l2));

 x = mouseX - 300;
 y = mouseY - 300;
 d = sqrt(x*x + y*y);
 //test sur la faisabilite
if (d<l1+l2){
 tht = atan(y/x);
 a = acos((d*d+l1*l1-l2*l2)/d/l1/2);
 b = acos((l1*l1+l2*l2-d*d)/l1/l2/2);
 th1=a+tht;
 th2=PI-b;
 if (x<0) {th1 += PI;}
 pushMatrix();
 translate(300,300);
 rotate(th1);
 stroke(255, 100, 0);
 strokeWeight(1);
 ellipse(0,0,2*(l1),2*(l1));
 strokeWeight(10);
 line(0,0,l1,0);
 translate(l1,0);
 stroke(100, 255, 0);
 strokeWeight(1);
 ellipse(0,0,2*l2,2*l2);
 rotate(2*PI-th2);
 strokeWeight(10);
 line(0,0,l2,0);
 popMatrix();
 }
}
```

Une librairie de gestion de la physique : jbox2d

VOIR LE TUTORIEL DE D. SHIFFMAN

http://natureofcode.com/book/chapter-5-physics-libraries/

hors sujet mais à voir aussi parce que c'est beau :
http://roberthodgin.com