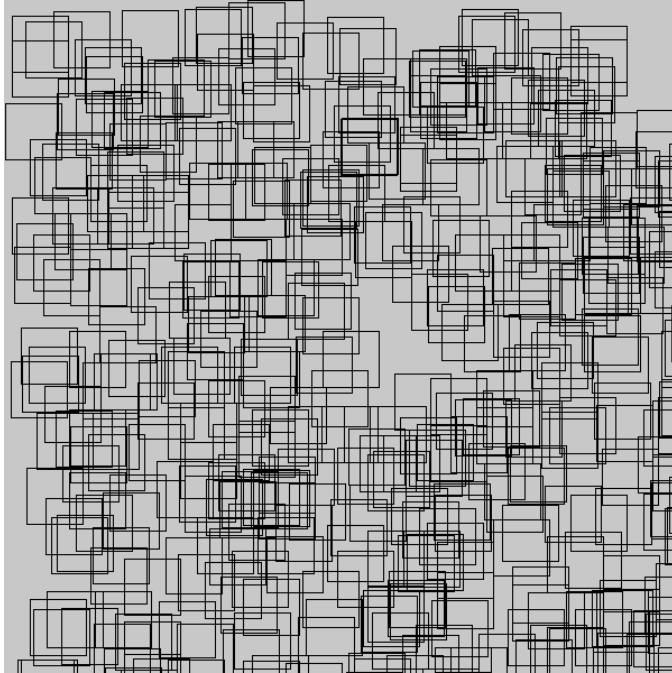


Médias interactifs numériques
synthèse
Processing & l'animation

Pierre Cubaud
Département d'informatique
CNAM

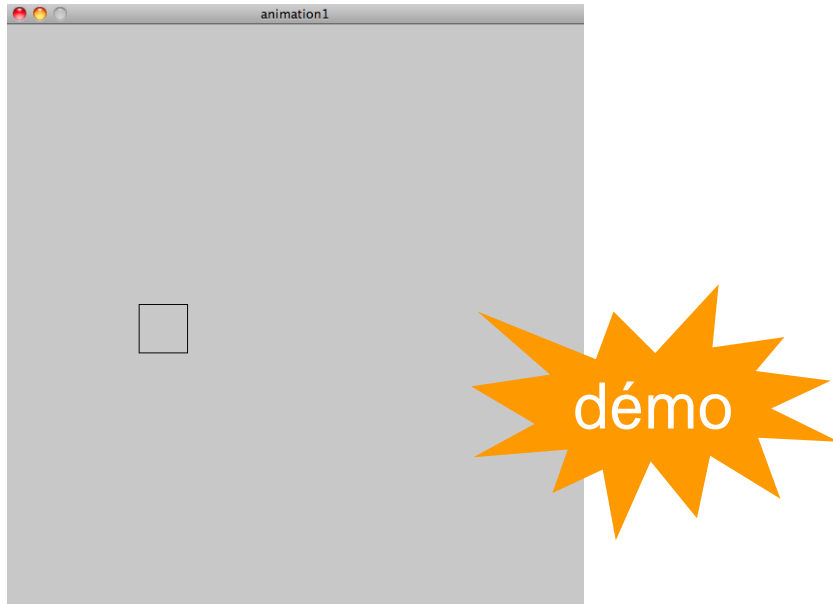
reprise du premier draw() fait en 



```
modereactif1
void setup() {
  size(600,600);
  noFill();stroke(0);
  background(200);
  frameRate(10);
}

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

mettre background() dans la boucle draw() :



pourquoi voit-on en fait plusieurs carrés ??

une première animation : le mouvement brownien

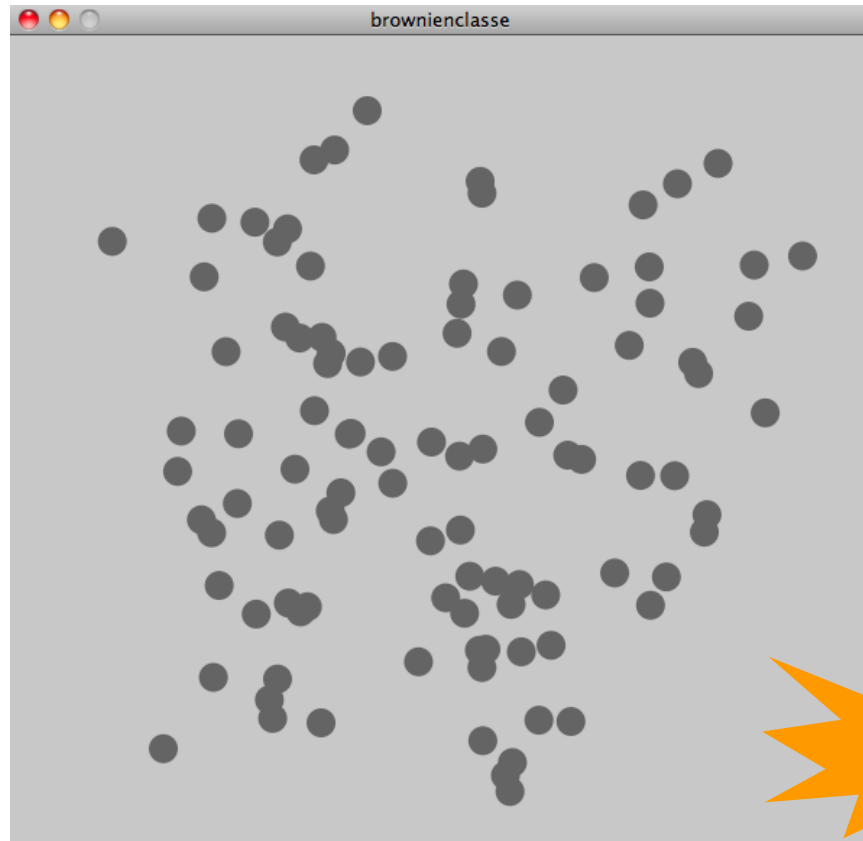
animation2

```
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();
}

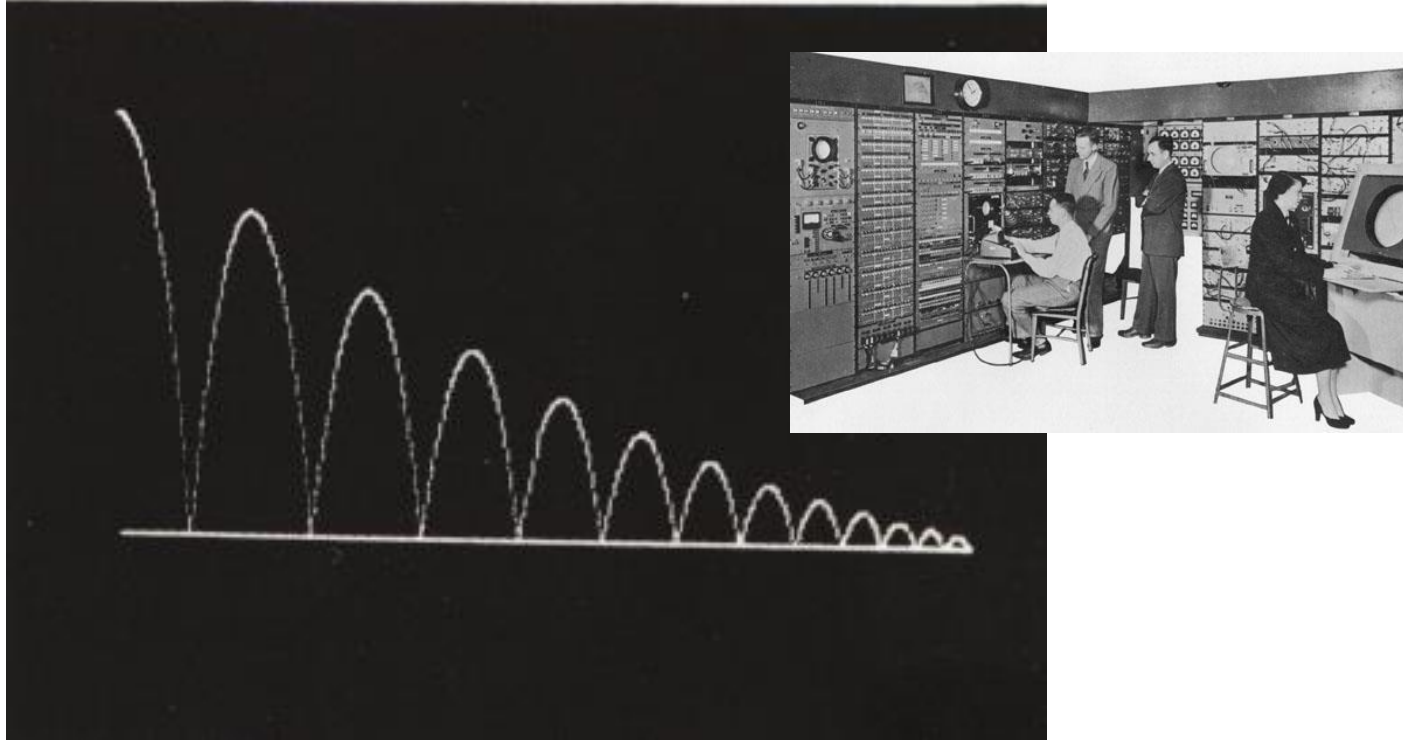
void draw() {
  background(0);
  for (int i=0;i<p.length;i++){
    p[i].dessin();
    p[i].evolution();
  }
}
```

```
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()

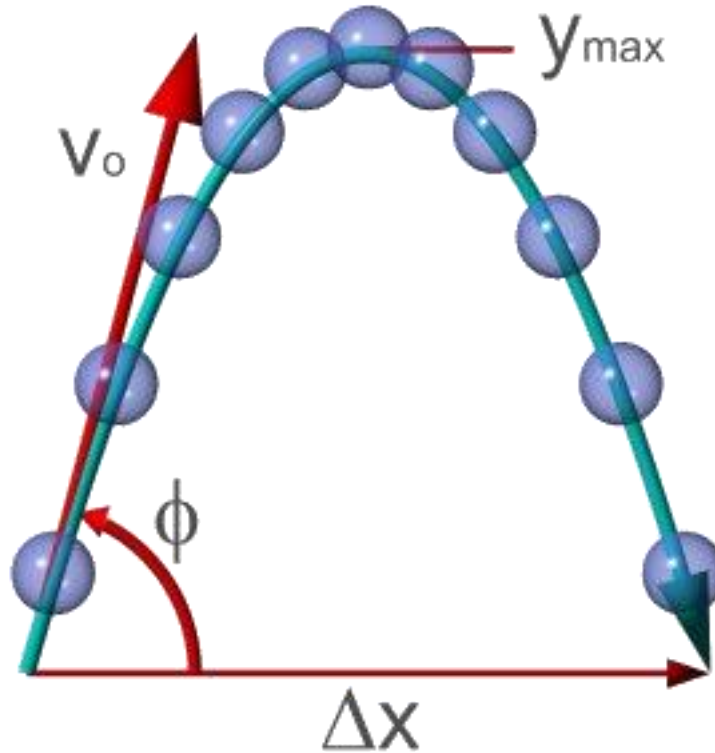
Deuxième exemple : le boulet de canon (dans le vide)



1949-51 - Projet WHIRLWIND

Un peu de physique :

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



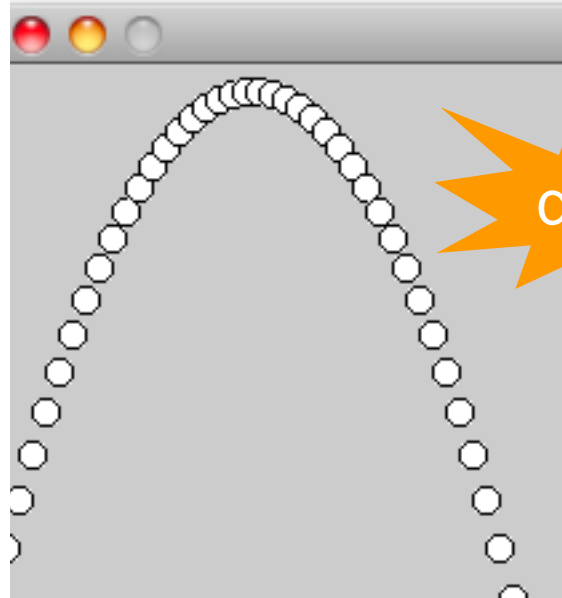
$$\sum \vec{F} = \begin{pmatrix} 0 \\ -mg \\ 0 \end{pmatrix} = m\vec{a} = \begin{pmatrix} m\ddot{x} \\ m\ddot{y} \\ m\ddot{z} \end{pmatrix} \Rightarrow \begin{cases} \dot{x} = v_0 \cos \Phi \\ \dot{y} = v_0 \sin \Phi - gt \\ \dot{z} = 0 \end{cases} \Rightarrow \begin{cases} x = v_0 t \cos \Phi \\ y =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

boulet

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

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

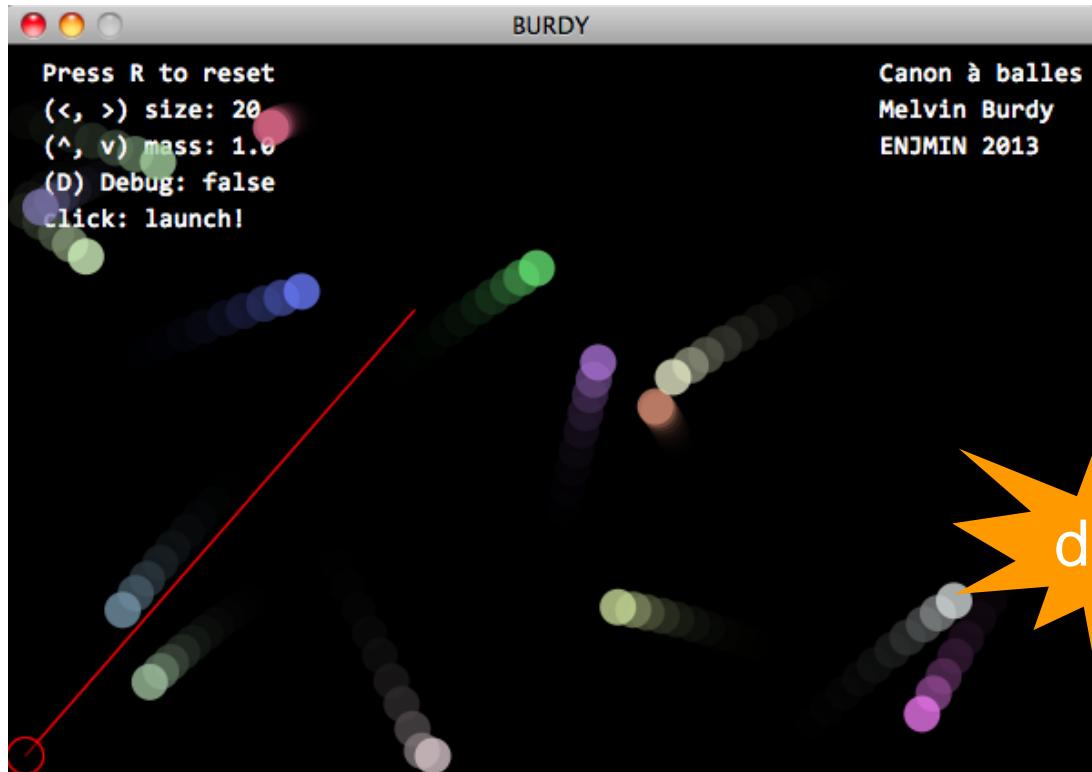
void draw() {
  ax = 0;
  ay = -1.0;
  vx = vx + ax;
  vy = vy + ay;
  x = x + vx;
  y = y + vy;
  ellipse(x,height-y,10,10);
}
```



démonstration

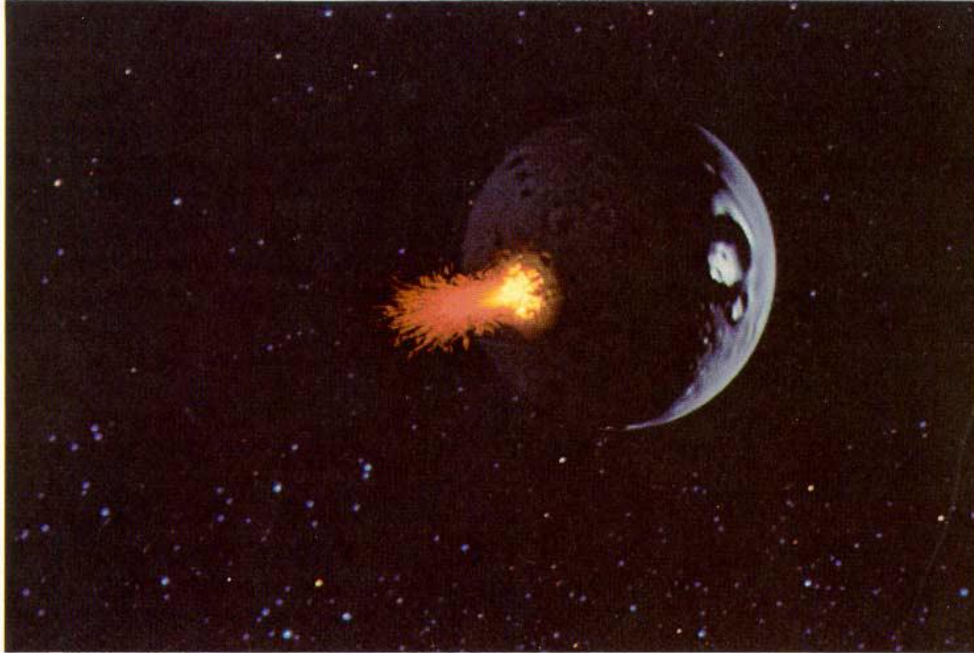
ajouter les
rebonds

Prise en compte des collisions entre objets

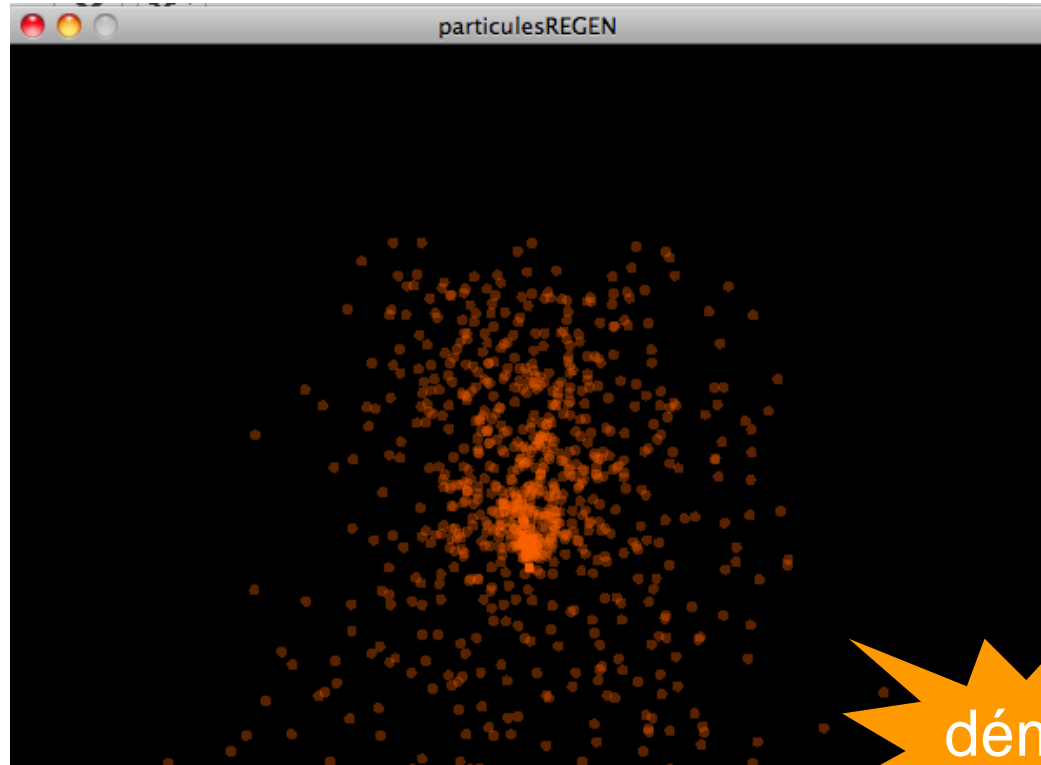


démo

Une extension intéressantes : les particules



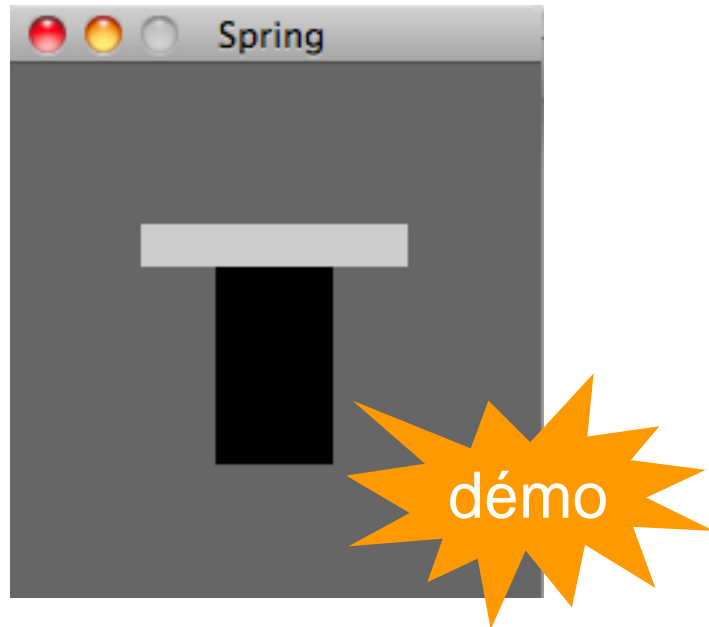
W. T. Reeves, "Particle Systems A Technique for Modeling a Class of Fuzzy Objects",
Computer Graphics, vol. 17, no. 3, pp 359-376, 1983



démo

Dernier exemple : 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 ?