M1 ENJMIN - MJV102

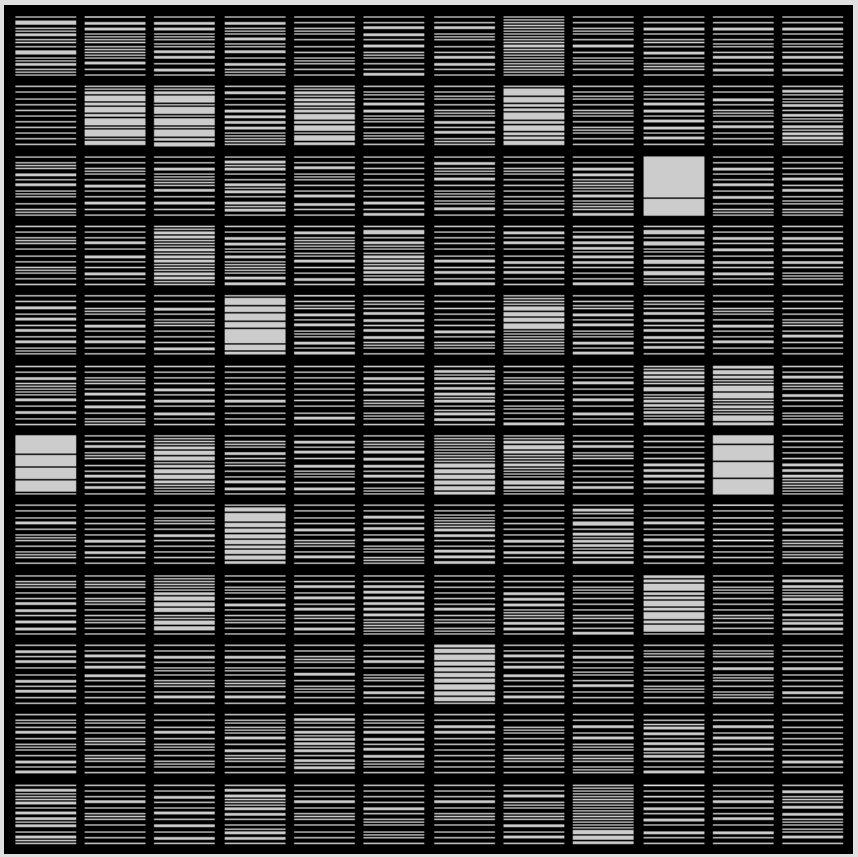
Conception et développement informatique

2 à 6 oct. 2017

Pierre Cubaud cubaud @ cnam.fr
Viviane Gal gal @ cnam.fr

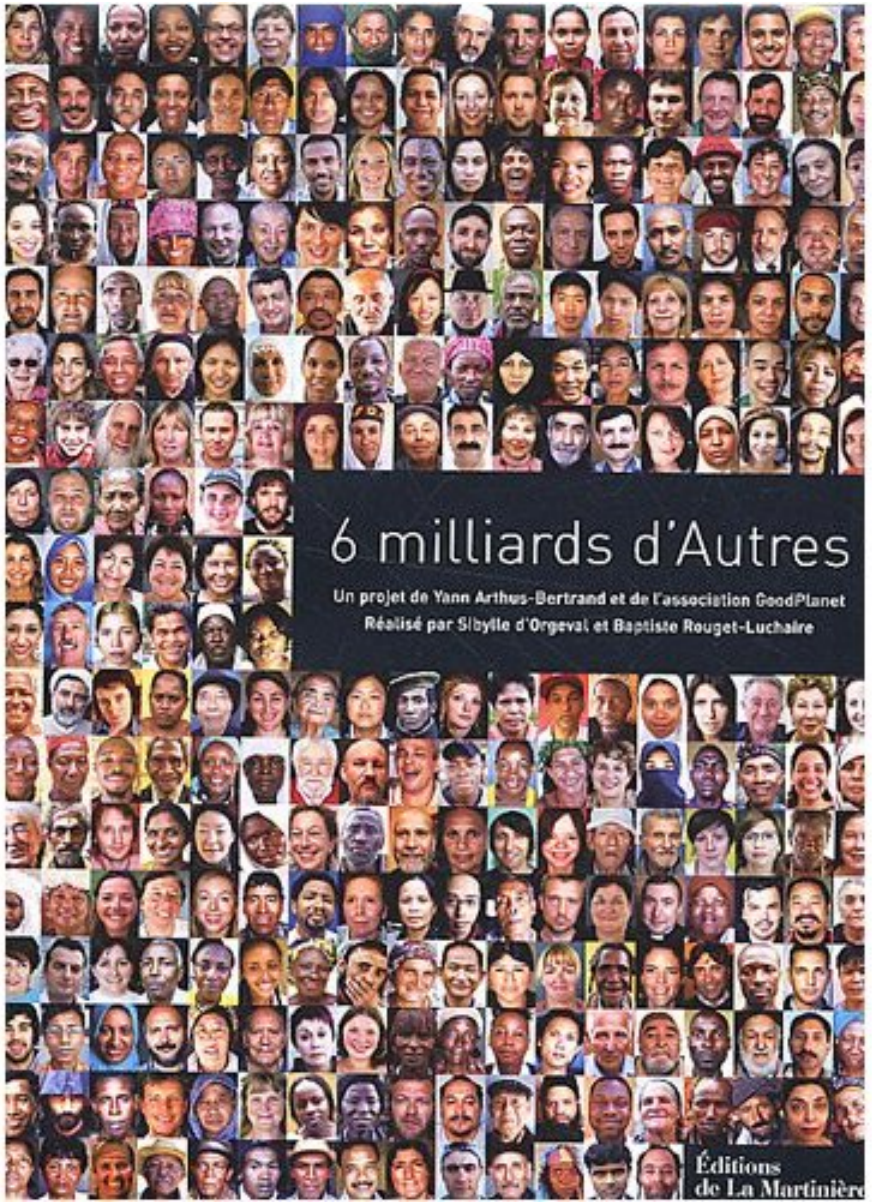
le cnam
École nationale du jeu et des médias interactifs numériques enjmin

JOUR 2. Itérations

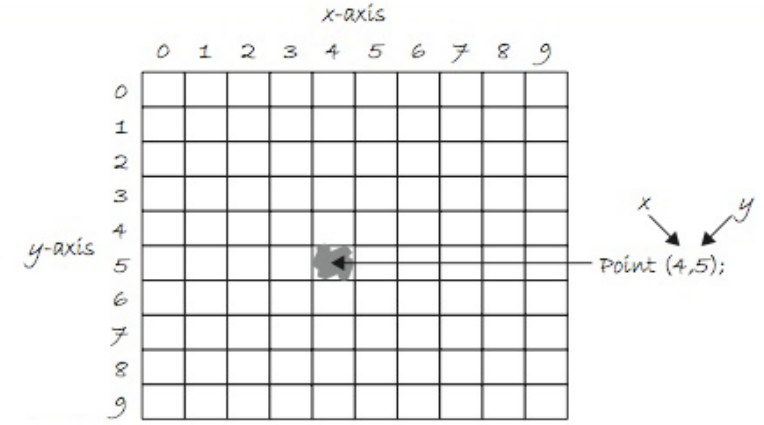


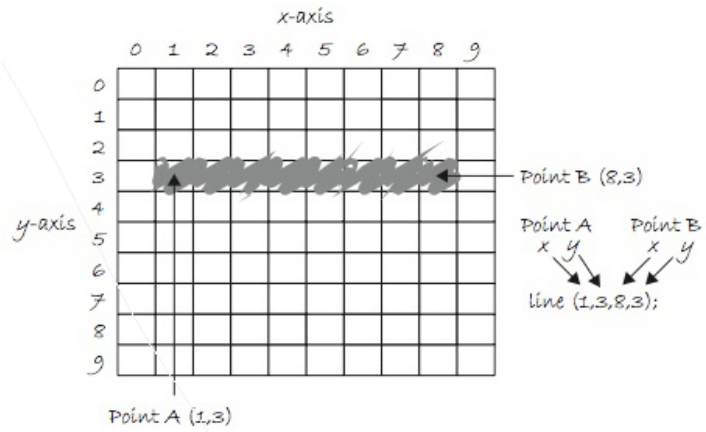
Reas&Fry, p. 152 - d'après Mark Wilson

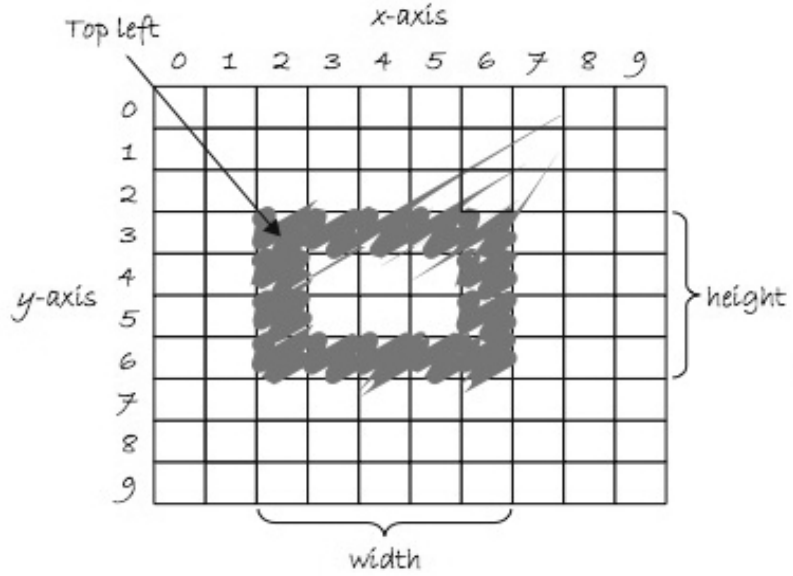
une esthétique de la répétition ?

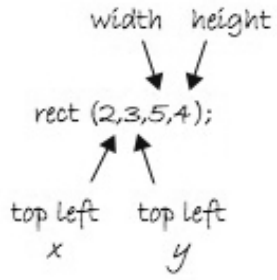


2.1. Formes élémentaires









point(…)
line(…)
rect(…)
ellipse(…)
arc(…)
triangle(…)
quad(…)

http://processing.org/learning/tutorials/basics/

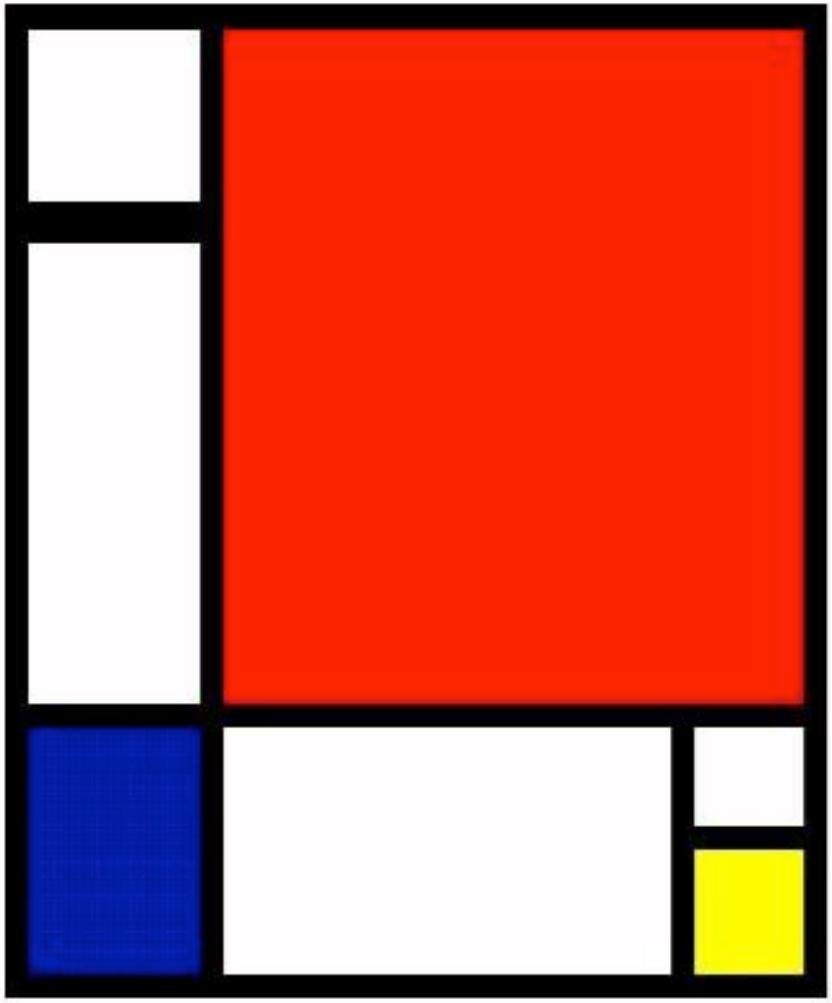
Les attributs de forme et de couleur

stroke(…) fill(…) noStroke() noFill()
strokeWeight(…) strokeCap(…) strokeJoin(…)

color(…) colorMode(…)
background(…)

```
background(0);
strokeWeight(64);
smooth();
stroke(255, 0, 0, 100); // rouge
point(47, 36);
stroke(0, 255, 0, 100); // vert
point(90, 47);
stroke(0, 0, 255, 100); // bleu
point(57, 79);
```

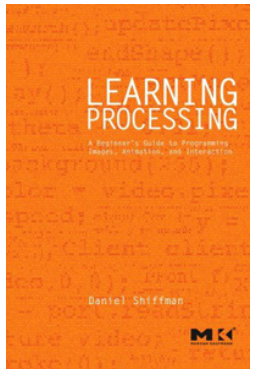
hommage à Mondrian



http://art.and.facts.site.free.fr/Site/artregions/images/stras/mondrian.jpg

```
size(400,500);

stroke(0);
strokeWeight(10);
strokeCap(SQUARE);
//le fond
background(255);
// le grand carre rouge
fill(255,0,0);
rect(100,0,300,300);
// le rectangle bleu
fill(0,0,255);
rect(0,300,100,200);
// le carre jaune
fill(255,255,0);
rect(300,400,100,100);
// le trait vertical pres du carré jaune
line(300,300,300,400);
// le trait horizontal en haut
line(0,100,100,100);
// le bord
noFill();
rect(0,0,400,500);

save("mondrian1.png");
```

(mondrian1.pde)



Zoog de Schiffman

```
size(640, 360);
background(255);
ellipseMode(CENTER);
rectMode(CENTER);

// Body
stroke(0);
fill(150);
rect(320, 190, 20, 100);

// Head
fill(255);
ellipse(320, 160, 60, 60);

// Eyes
fill(0);
ellipse(301, 160, 16, 32);
ellipse(339, 160, 16, 32);

// Legs
stroke(0);
line(310, 240, 300, 250);
line(330, 240, 340, 250);
```

(zoog.pde)

Les commentaires en Java

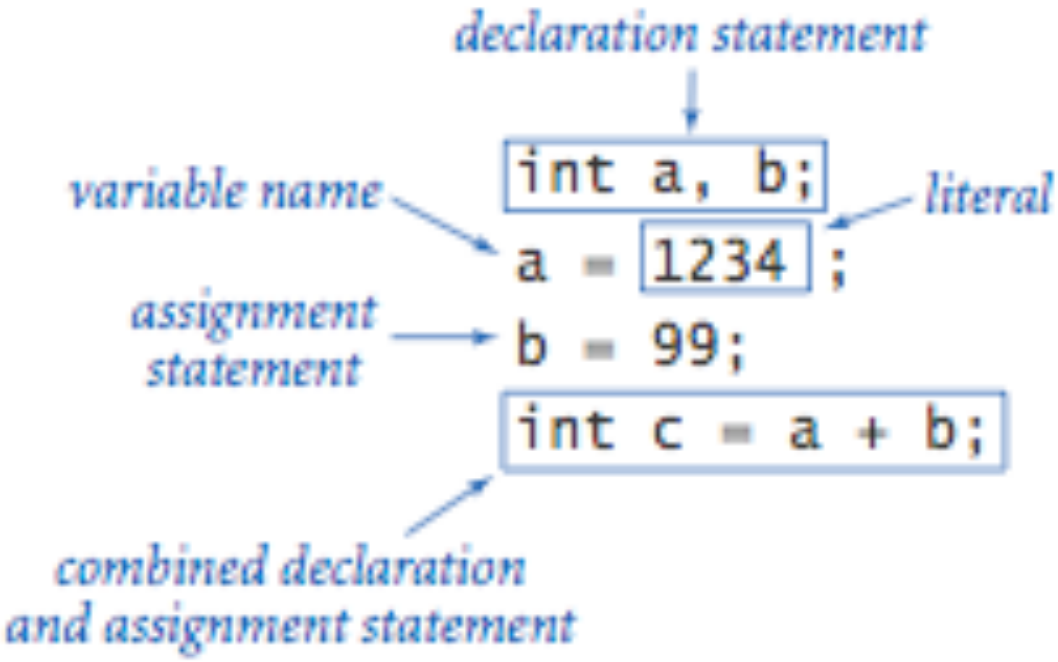
Il est très important de commenter ses programmes :
pour les autres ou pour soi-même plus tard

```
// commentaire sur une ligne

/*
 commentaires
 sur plusieurs lignes
*/

/**
 commentaire qui va etre reconnu par javadoc
*/
```

Il existe aussi des outils pour l'annotation (@author etc)

2.2 Les variables en Java

- doivent être déclarées avant d'être utilisées dans des calculs
- ont un type associé (nombre, texte, booléen) non modifiable
- n'ont pas de valeur initiale par défaut
- peuvent être des constantes (comme PI) : mot clé **final**



Les valeurs littérales en Java

- Un entier est représenté classiquement :

```
3       56       987
```

- Un réel est représenté avec la notation pointée :

```
3.1415
```

- Un caractère est représenté entre quotes pour le distinguer de tout autre identificateur :

```
'A'    '7'      '$'
```

- Une chaîne de caractères est représentée entre guillemets pour la distinguer de tout autre identificateur ou mot clé :

```
"Hello World "
```

- Un booléen est représenté par les symboles :

```
true, false
```

Exemple : Belle Marquise (Molière, Bourgeois gentilhomme)

bellemarquise

```
String a="Belle Marquise ";
String b="Vos beaux yeux ";
String c="Mourir d'amour ";
String d="Me font ";

println(a+b+c+d);
println(a+b+d+c);
println(a+c+b+d);
println(a+c+d+b);
println(a+d+b+c);
println(a+d+c+b);
```

```
Display 0 does not exist, using the default display instead.
Belle Marquise Vos beaux yeux Mourir d'amour Me font
Belle Marquise Vos beaux yeux Me font Mourir d'amour
Belle Marquise Mourir d'amour Vos beaux yeux Me font
Belle Marquise Mourir d'amour Me font Vos beaux yeux
Belle Marquise Me font Vos beaux yeux Mourir d'amour
Belle Marquise Me font Mourir d'amour Vos beaux yeux
```

- "additionner" des chaines = mettre bout à bout = concaténer
- combien d'autres permutations ?

Identifiants autorisés en Java

- pas que pour les variables (traitements, classes etc)
- sensible à la casse ≠ CASSE ≠ CaSsE
- on peut utiliser le $ et le _ (underscore)
- on peut utiliser les accents éàç (en Unicode) : bof

Illégal : 123xp légal : a, a, xp_12, _$

Coutumes avec la casse :

data : une variable mineure, un package
Hello : une classe, un type
lireSuivant : une "grosse" variable, un traitement (méthode)
MAXWIDTH : une constante

Mauvais style :

ceciestunnomtroplongdevariable xp12

Mots reservés de Java

abstract	continue	for	new	switch
assert[Note 1]	default	goto[Note 2]	package	synchronized
boolean	do	if	private	this
break	double	implements	protected	throw
byte	else	import	public	throws
case	enum[Note 3]	instanceof	return	transient
catch	extends	int	short	try
char	final	interface	static	void
class	finally	long	strictfp[Note 4]	volatile
const[Note 2]	float	native	super	while

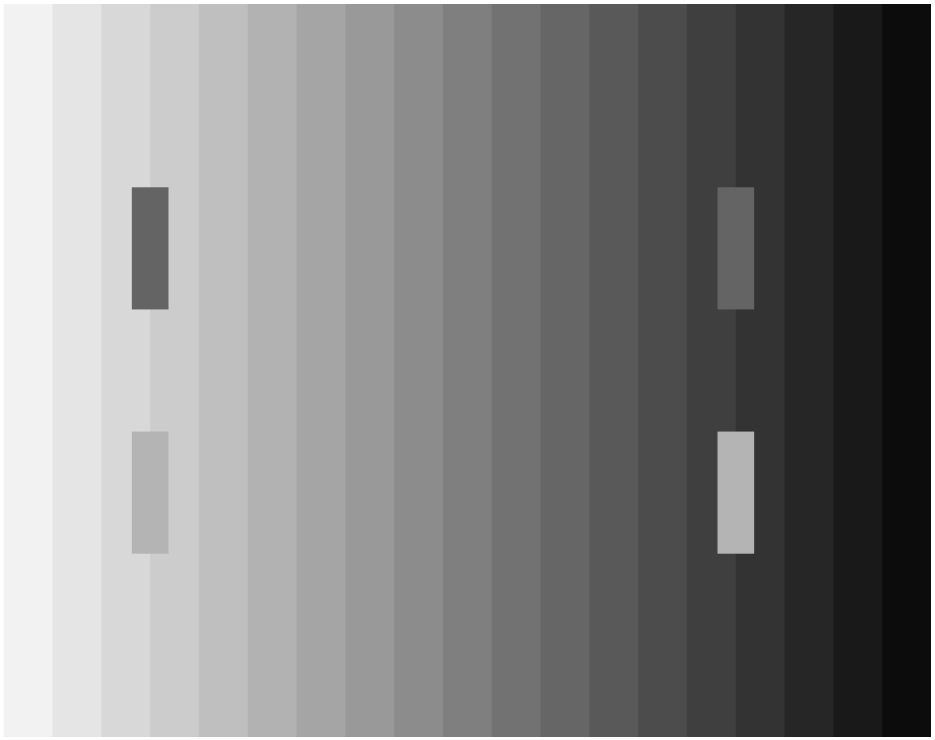
1. ^ Keyword was introduced in J2SE 1.4
2. ^ [a] [b] Keyword is not used
3. ^ Keyword was introduced in J2SE 5.0
4. ^ Keyword was introduced in J2SE 1.2

(wikipedia)

Rque : attention aussi à ceux de Processing

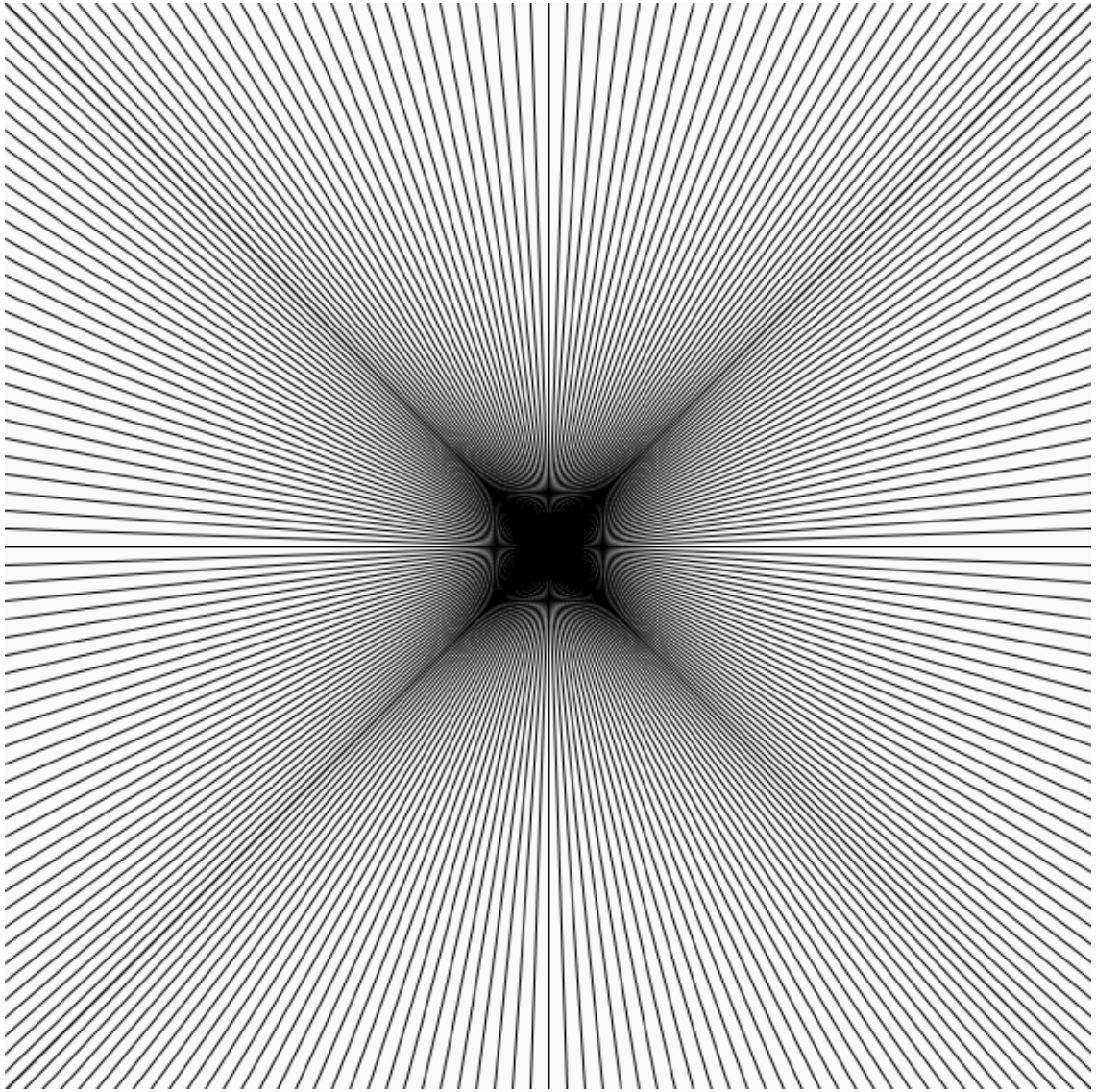
2.3. Les boucles

```
for (int i=0; i<1234567; i++) {
 println(i);
}
```



```
int N=20;
size(800,600);
noStroke();
for (int i=0; i<N;i++){
  fill(map(i,0,N,255,0));
  rect(i*width/float(N),0,width/N,height);
}
rectMode(CENTER);
fill(100);
rect(160,200,30,100);
rect(640,200,30,100);
fill(180);
rect(160,400,30,100);
rect(640,400,30,100);
save("contraste.png");
```

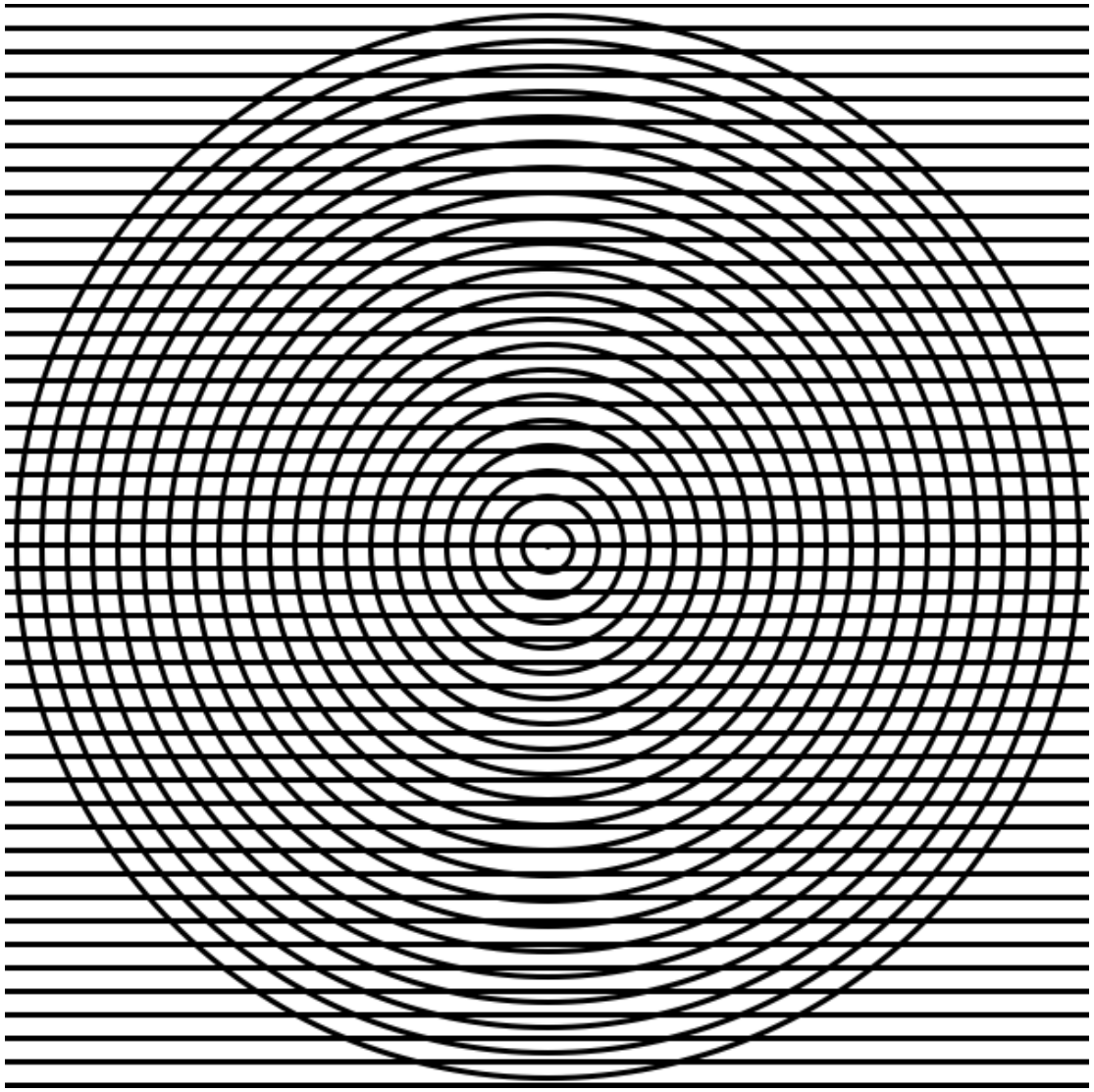
(contraste.pde)



```
size(600,600);
background(250);
for (int i=0;i<=600;i+=6){
 line(i,0,600-i,600);
 line(0,i,600,600-i);
}
save("alias.png");
```

(aliasdroite.pde)

Aute exemple de moiré



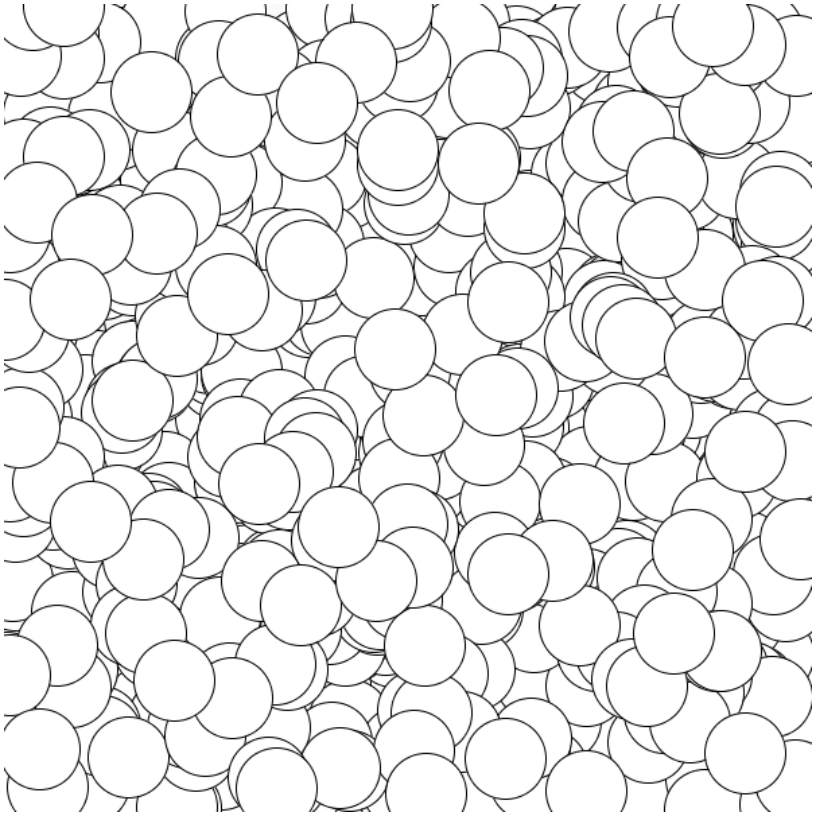
```
int delta = 13;

size(600,600);
strokeWeight(3);
stroke(0); noFill(); smooth();
background(255);

////// dessin grille
int y = 0;
while (y<height){
 line(0,y,width,y);
 y += delta;
}
/////// dessin cercles
int r = 0;
while (r<height/2){
 ellipse(width/2,height/2,2*r,2*r);
 r += delta+1; ///// essayer aussi avec autre valeur
}

save("moiresimple.png");
```

(moiresimple.pde)

2.4. La perturbation

Bridget Riley - Circles

random(min, max)

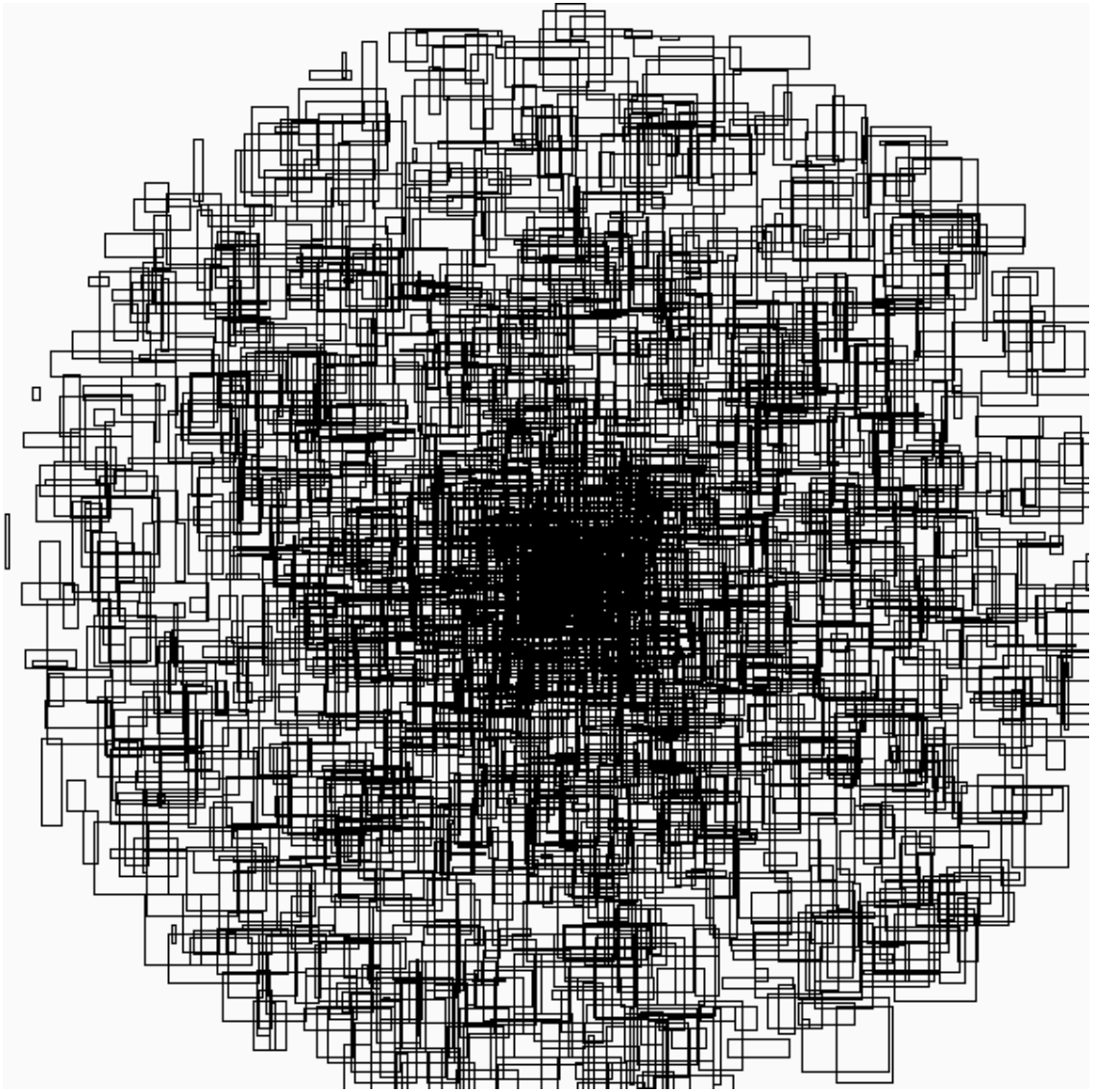


```
size(600,600);

background(250);
//noStroke();fill(0,0,128,30);

for (int i=0; i<1000; i++) {

 float x=random(0,600);
 float y=random(0,600);

 ellipse(x,y,60,60);
}

save("generatif1.png");
```

(generatif1.pde)

modifier :

- les attributs (stroke, fill)
- le nombre d'itérations
- la taille des cercles



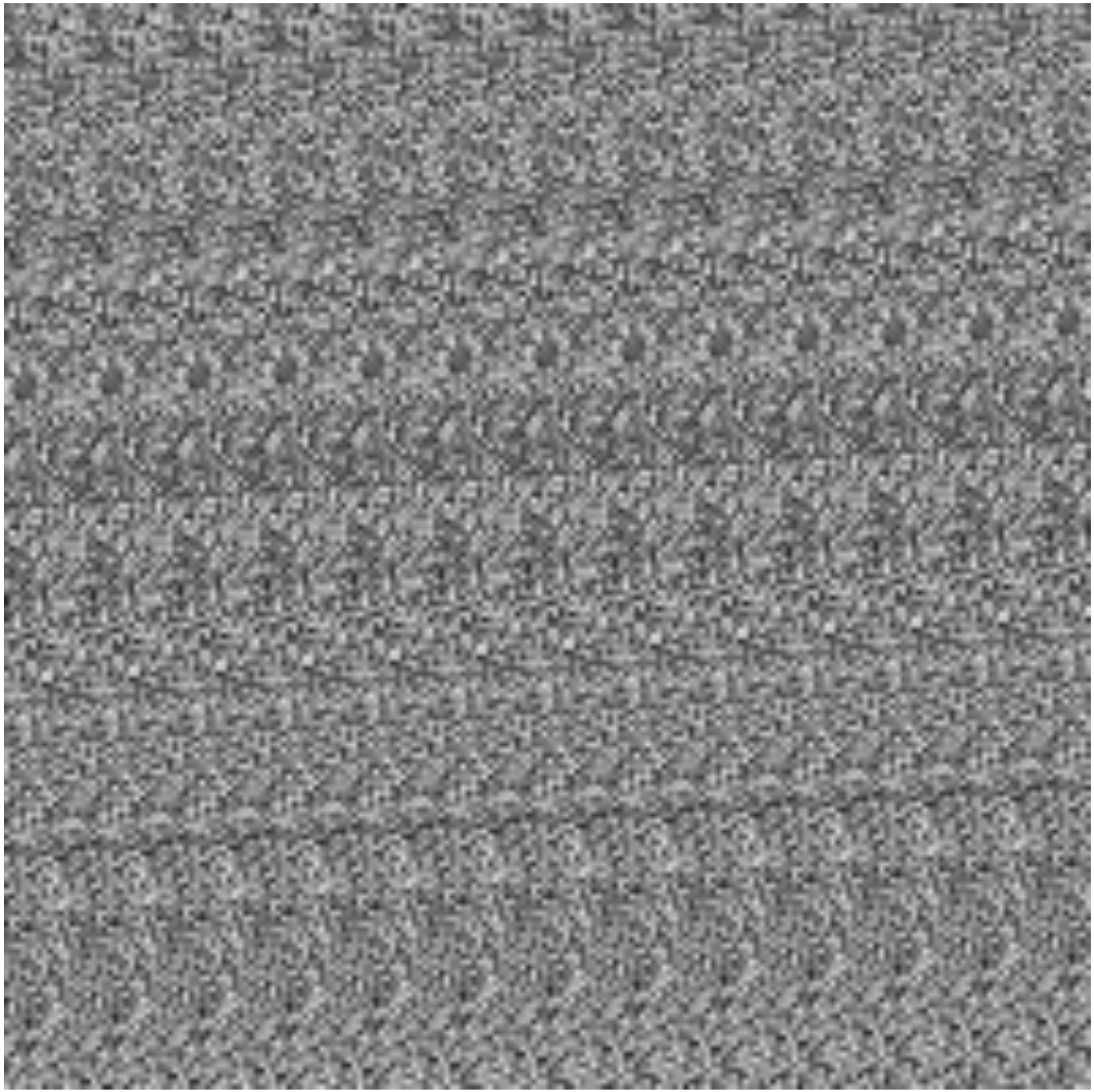

5,5t

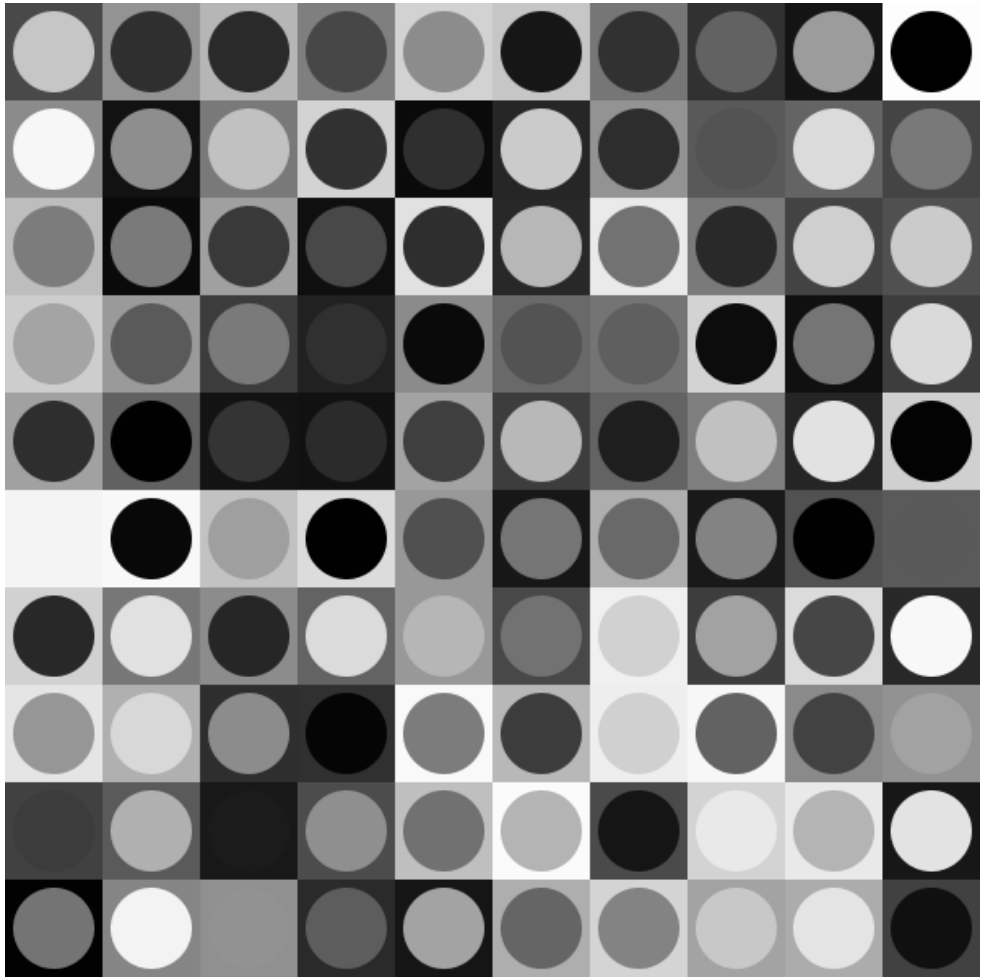
```
float r,th,x,y,l,h;

size(600,600);
background(250); noFill(); stroke(0);

for (int i=0; i<2000; i++) {
  r=random(0,300);
  th=random(0,2*PI);
  x=300+r*cos(th);
  y=300+r*sin(th);
  l=random(1,50);
  h=random(1,50);
  rect(x,y,l,h);
}

save("mondrian2.png");
```

(mondrian2.pde)

« To paint well is simply this: to put the right color in the right place. »
Paul Klee

```
for (int x = 0; x < width; x++) {
    for (int y = 0; y < height; y++) {
        stroke( 255*noise(x,y));
        point(x,y);
    }
}
```





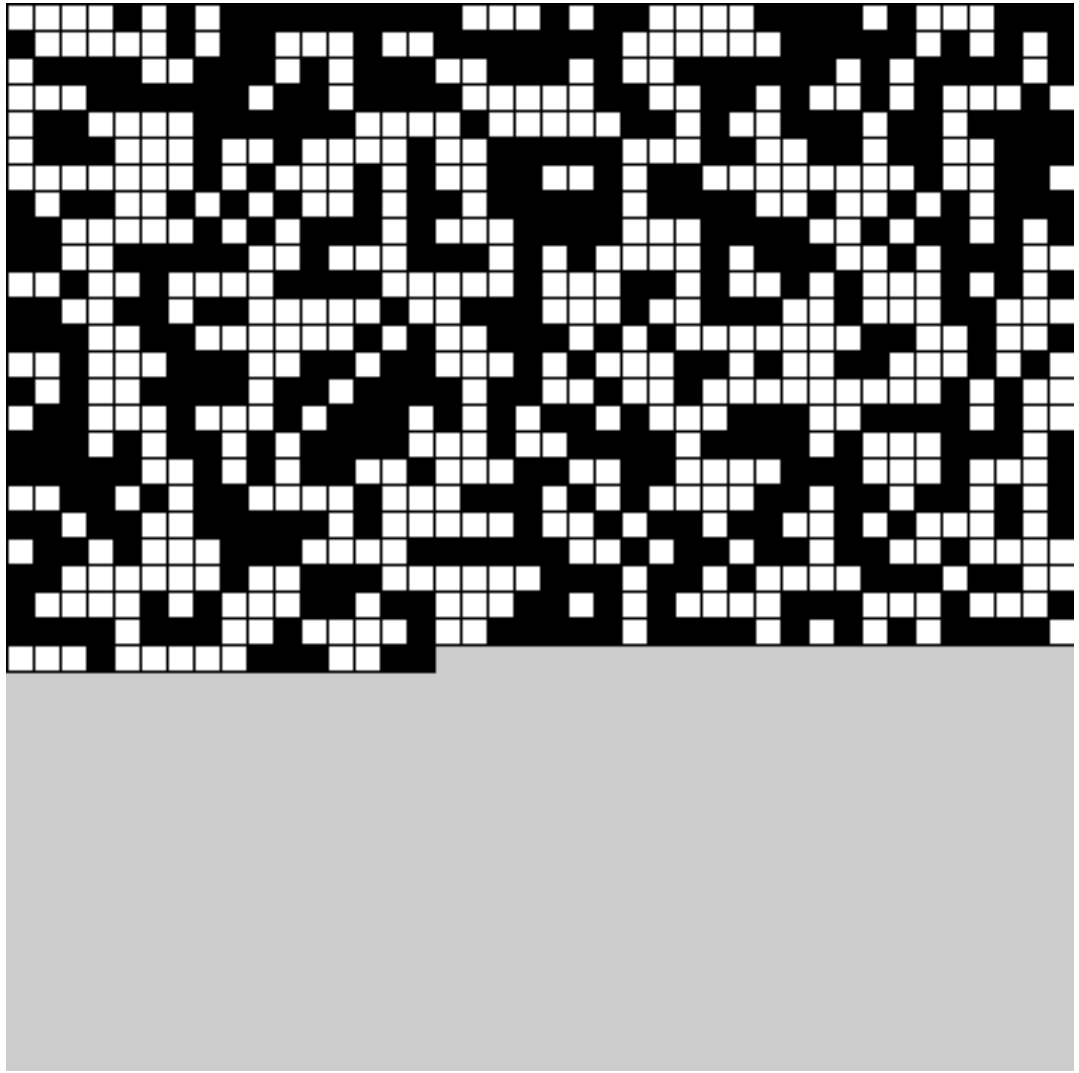
Hommage à Le Parc et Vasarely

```
size(600,600);
rectMode(CENTER);
ellipseMode(CENTER);
smooth();
noStroke();

for (int x=30; x<600; x+=60) {
  for (int y=30; y<600; y+=60) {
    fill(random(0,255));
    rect(x,y,60,60);
    fill(random(0,255));
    ellipse(x,y,50,50);
  }
}

save("leparc.png");
```

(leparc.pde)

2.6. La décision

```
if ( score < 314) {
 println(« game over »);
}
else {
 println(« shoot again »);
}
```



```
size(400,400);
int x=0;
int y=0;
for (int i=0;i<1000;i++){
  float t=random(0,1);
  if (t<0.5)
   fill(0);
  else
   fill(255);
  rect(x,y,10,10);
  if (x < width) x+=10;
  else {
    x=0;
    if (y < height) y+=10;
    else y=0;
  }
}
save("pavagernd.png");
```

(pavagernd.pde)

Le grand mystère de la suite de Syracuse

$$\forall n > 1,\ A_{n+1} = \begin{cases} 3A_n + 1 & \text{si } A_n \text{ impair} \\ A_n / 2 & \text{sinon} \end{cases}$$

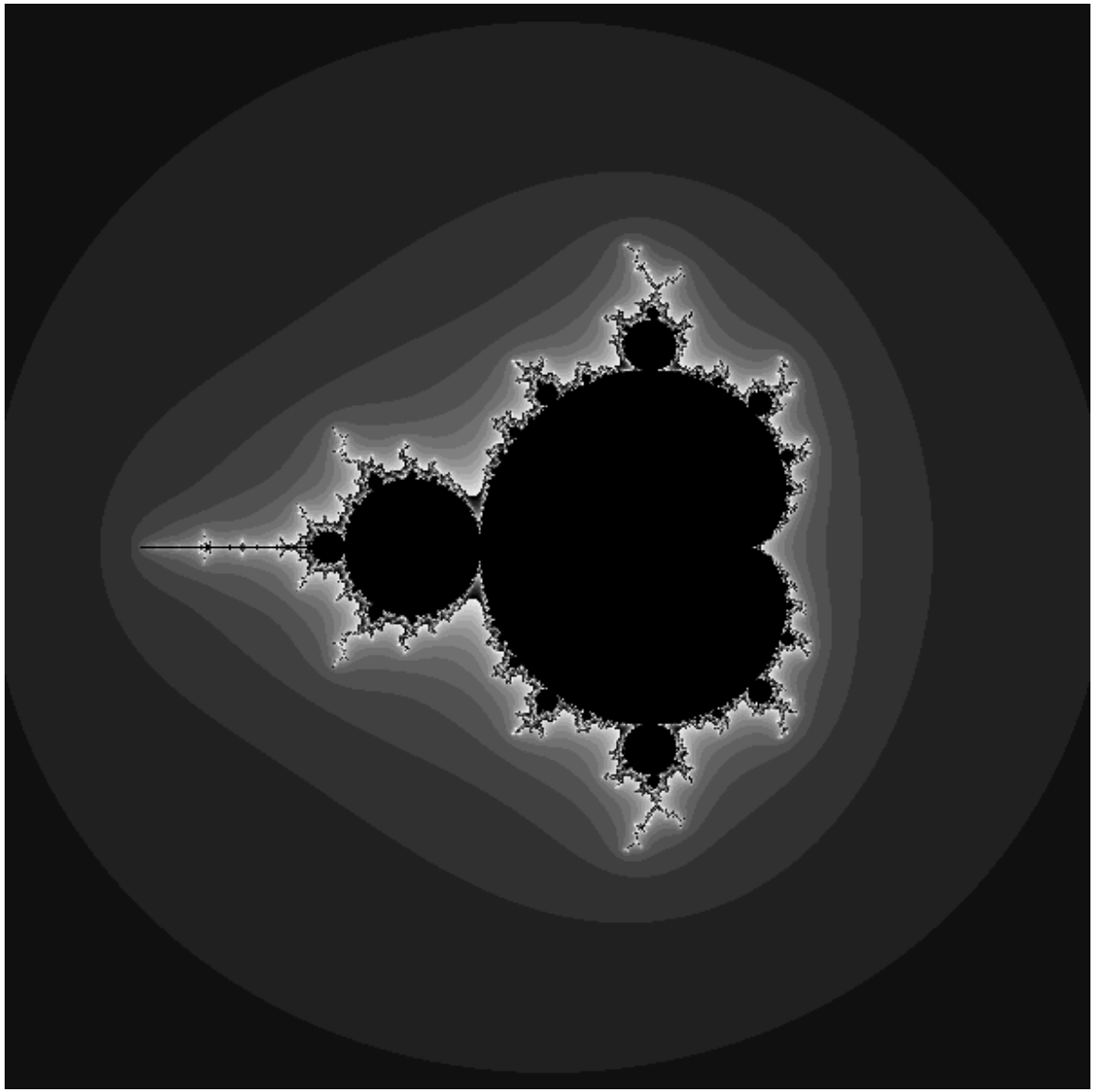
Quelque soit la valeur initiale de A, la suite se termine en bouclant infiniment sur les valeurs 4, 2, 1, 4.

A ce jour, personne n'a pu démontrer pourquoi, ni prédire au bout de combien de termes la boucle est atteinte.

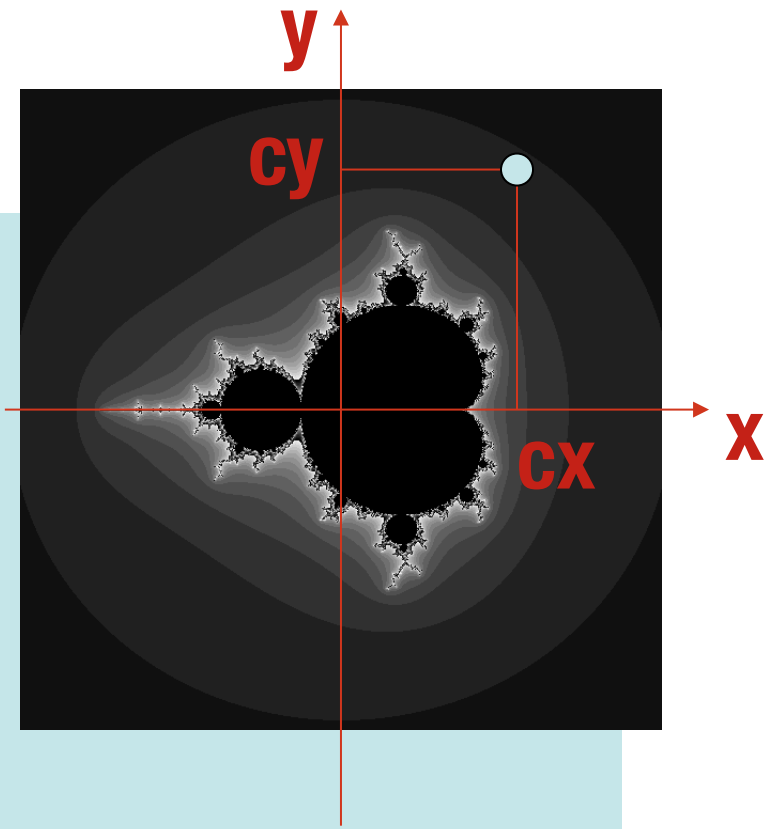
```
int c=1;
int A=27;
print(A);
while (A != 1) {
 if (A%2 == 0)
   A = A/2;
 else
  A = 3*A + 1;
 c++;
}
println(":"+c);
```

(troisaplus1.pde)

L’ensemble de Mandelbrot



etude de Z(n+1) = Z(n)^2+c
avec Z(0) = c (complexe)




```
for (...x...) for (...y...) {
  float cx = map(x,0,width,-2.5,1.5);
  float cy = map(y,0,height,2,-2);
  int n = 0;
  float a=cx;
  float b=cy;
  while (n < 200) {
    float aa = a * a;
    float bb = b * b;
    float twoab = 2.0 * a * b;
    a = aa - bb + cx;
    b = twoab + cy;
    // test de divergence : on se limite à norme(Z) > 16
    if (aa + bb > 16.0f) {break;}
    n++;
   }
  if (n == 200) stroke(0); else stroke(n*16 % 255);
}}
```

Comment en faire une bonne interface zoomable ?

beginShape() … endShape()

To make a sharp turn, use the same position to specify the vertex and the following control point. To close the shape, use the same position to specify the first and last vertex.

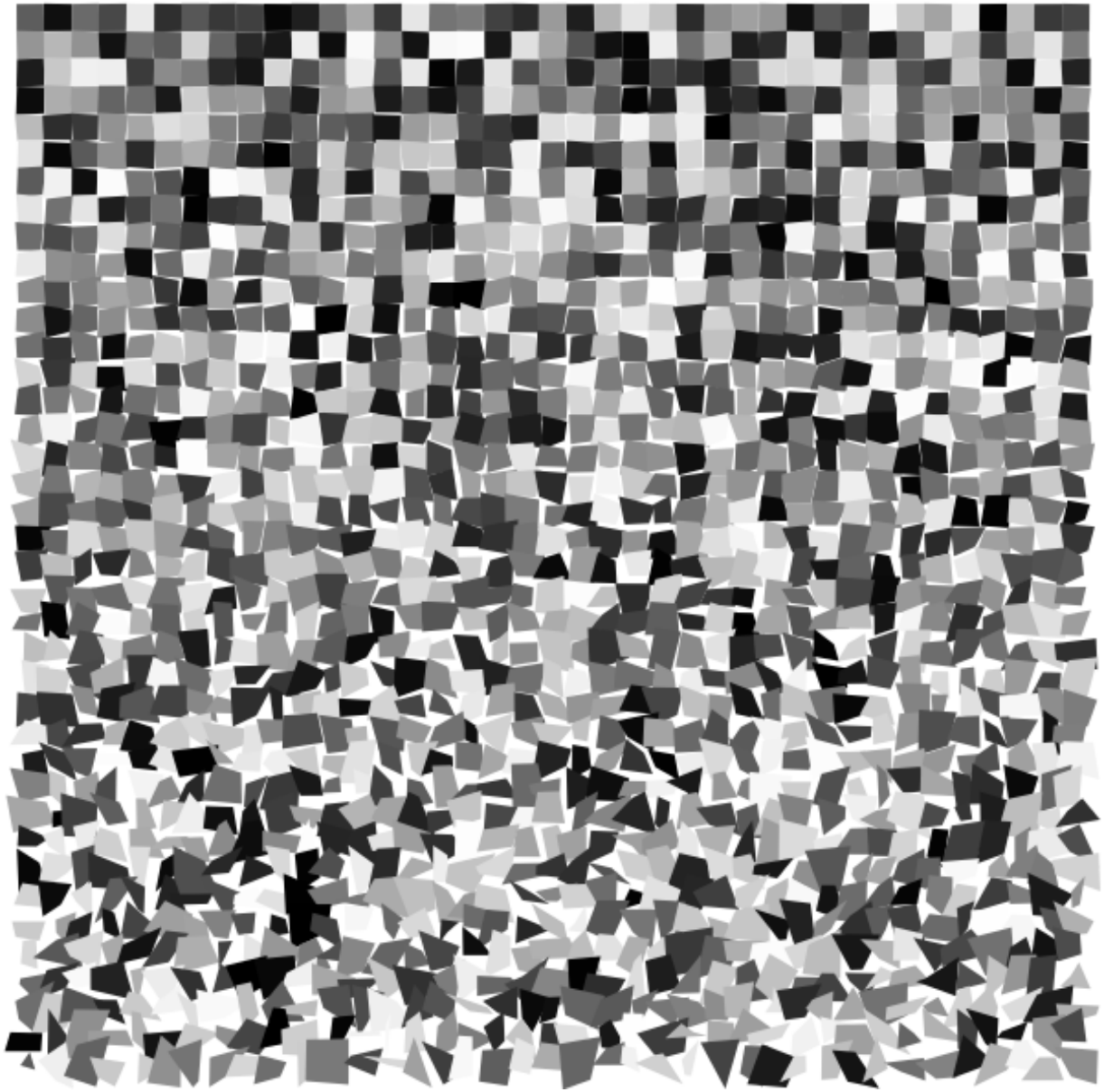
```
smooth();
noStroke();
beginShape();
vertex(90, 39);                          // V1 (see p.76)
bezierVertex(90, 39, 54, 17, 26, 83);    // C1, C2, V2
bezierVertex(26, 83, 90, 107, 90, 39);   // C3, C4, V3
endShape();
```

Place the `vertex()` function within `bezierVertex()` functions to break the sequence of curves and draw a straight line.

```
smooth();
noFill();
beginShape();
vertex(15, 40);                          // V1 (see p.76)
bezierVertex(5, 0, 80, 0, 50, 55);       // C1, C2, V2
vertex(30, 45);                          // V3
vertex(25, 75);                          // V4
bezierVertex(50, 70, 75, 90, 80, 70);    // C3, C4, V5
endShape();
```

A good technique for creating complex shapes with `beginShape()` and `endShape()` is to draw them first in a vector drawing program such as Inkscape or Illustrator. The coordinates can be read as numbers in this environment and then used in Processing. Another strategy for drawing intricate shapes is to create them in a vector-drawing program and then import the coordinates as a file. Processing includes a simple library for reading SVG files. Other libraries that support more formats and greater complexity can be found on the Processing website at *www.processing.org/reference/libraries*.

Reas & Fry p. 77



Quads - Greenberg, p. ???

(monquads.pde)

```
int DIM=15;
float VAR=0.2;

size(600,600);
background(255);
smooth();
noStroke();

int k = 0;
for (int i=0;i<height; i+=DIM){
  for (int j=0; j<width; j+=DIM){
    fill(random(0,255));
    float x0 = 0 + random(-k*VAR, k*VAR);
    float y0 = 0 + random(-k*VAR, k*VAR);
    float x1 = DIM + random(-k*VAR, k*VAR);
    float y1 = 0 + random(-k*VAR, k*VAR);
    float x2 = DIM + random(-k*VAR, k*VAR);
    float y2 = DIM + random(-k*VAR, k*VAR);
    float x3 = 0 + random(-k*VAR, k*VAR);
    float y3 = DIM + random(-k*VAR, k*VAR);
    quad(x0,y0,x1,y1,x2,y2,x3,y3);
    translate(DIM, 0);
  }
  translate(-width,DIM);
  k ++;
}
save("quads.tif");
```

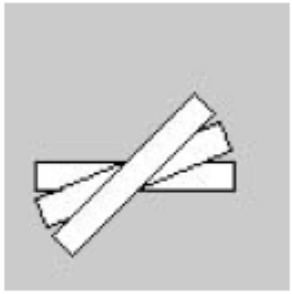
Le repère de tracé (la caméra)

pushMatrix() popMatrix() translate(…) scale(…) rotate(…)

```
translate(width/2, height/2);
rotate(PI/8);
rect(-25, -25, 50, 50);
```

```
rotate(PI/8);
translate(width/2, height/2);
rect(-25, -25, 50, 50);
```

Casey & Reas
p.139 et 141



```
translate(45, 60);
rect(-35, -5, 70, 10);
rotate(-PI/8);
rect(-35, -5, 70, 10);
rotate(-PI/8);
rect(-35, -5, 70, 10);
```

ex: création de trame par déplacement du repère


```
for (int i=0; i<24; i++){
   pushMatrix();
   for (int j=0; j<12; j++){
      ellipse(0,0,60,60);
      translate(25,0);
   }
   popMatrix();
   translate(0,25);
 }
```