CNAM spécialité informatique — NSY116 Multimédia et IHM

ED1. Premiers programmes avec Processing

P. Cubaud < cubaud @ cnam.fr > oct. 2008

###CORRIGE###

Exercice 1. Boites et cercles

```
// NSY116 - ED1 - premiers exemples processing
// on essaie ici de recreer un tableau de Mondrian
// P. Cubaud oct 2008
//

size(400,460);
smooth();
stroke(0);
background(255);
strokeWeight(7);
strokeCap(SQUARE);

// le grand carre rouge
fill(255,0,0);
rect(100,0,300,300);
// le rectangle bleu
fill(0,0,255);
rect(0,300,width-300,height-300);
// le carre jaune
int c = (height-300)/ 2;
fill(255,255,0);
rect(width-c,height-c,c,c);
// le trait vertical pres du carré jaune
line(width-c,height-c,width-c,300);
// le bord du tableau
strokeWeight(10);
noFill();
rect(0,0,width,height);
// le trait horizontal en haut
line(0,100,100,100);

save("mondrian1.png");


/// premiere generation procedurale (code auditeur 2007-8)

int NBR= 3000;
int winSize= 600;
int x,y,rectWith,rectHeight;
float a;
```

```
size(winSize, winSize, P3D);
int i= 0;
int maxSize= int(pow(winSize / 2, 2));
noStroke();
background(255,255,255);
while (i < NBR){
  x = int(random(winSize));
  y = int(random(winSize));
  if ((pow(x - winSize / 2,2) + pow(y- winSize / 2,2)) < maxSize)
  {
    rectWith = int(random(15));
    rectHeight = int(random(20));
    fill(int(random(255)), int(random(255)), int(random(255)));
    rect(x,y,rectWith,rectHeight);
    ++i;
  }
}
```

on devrait faire mieux (c'est à dire éviter le test d'appartenance du point à l'intérieur d'un cercle) en tirant au sort une distance D par rapport au centre et un angle ALPHA puis calculer X = WIDTH/2 + D*COS(ALPHA) et Y = HEIGHT/2 + D*SIN(ALPHA)

```
//// hommage à Bridget Riley

size(1200,400);
smooth();
noFill();
background(255);
stroke(0);

int D = int(height/5.0);
int x=0;
int y=0;
for (int n=1;n<=230;n++){
 if (random(1)>0.35) ellipse(x,y,D,D);
 x += D/2;
 if (x>width) {
   y += D/2;
   x = 0;
 }
}
save("rileycircles.png");
```

Exercice 2. Symétries

```
boolean dessin = false;

void setup() {
  size(500, 500);
  background(226);
  smooth();
  strokeWeight(20);
  stroke(0, 100);
}

void draw() {
 if (dessin){
  point(mouseX,mouseY);
  point((width-mouseX)%width,mouseY);
// si on veut 4 points, ajouter les deux lignes ci-dessous
//  point(mouseX,(height-mouseY)%height);
//  point((width-mouseX)%width,(height-mouseY)%height);
 }
}

void mousePressed() {
  dessin = false;
}

void mouseReleased() {
  dessin = true;
}
```