UNIVERSITÉ PARIS SUD

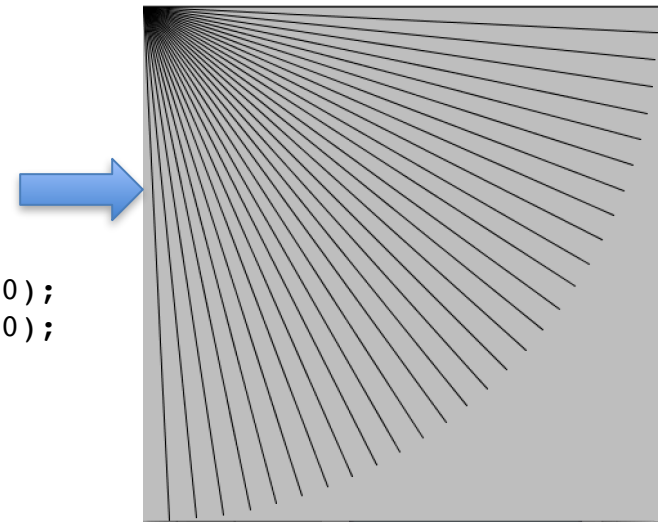
Licence MPI. Info 112 (IGOR)
F. Vernier, C. Jacquemin

Examen 2017-2018, session 1 (2h, barème /20)

1. Dessin (6 pts)

Dans cet exercice, il vous est demandé de fournir un certain nombre d'éléments nécessaires à la réalisation d'une application écrite en langage Processing. Nous partirons du programme ci-dessous qui donne le résultat ci-contre. (Un corrigé sera distribué à la toute fin des 2 heures)

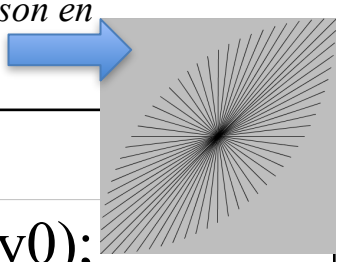
```
1.  void setup(){
2.    size(400, 400);
3.  }
4.
5.  void draw(){
6.    background(192);
7.    strokeWeight(1);
8.    for (int i=0; i<30; i++){
9.      float x0 = 0 + 400*cos(i*PI/2/30);
10.     float y0 = 0 + 400*sin(i*PI/2/30);
11.     float x1 = 0;
12.     float y1 = 0;
13.     line (x1, y1, x0, y0);
14.   }
15. }
```



1.1. ***(1pt)*** *Comment s'appelle le principe de math utilisé aux lignes 9 et 10 ?*

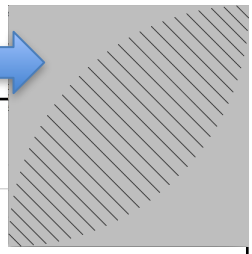
Trigonometrie

1.2. ***(2 pts)*** *Changer 2 lignes afin de faire une étoile en forme de calisson en changeant les coordonnées x1,y1*



```
line (width/2, height/2, x0, y0);
line (width/2, height/2, width-x0, height-y0);
```

1.2. ***(2 pts)*** *Comment relier les points 2 à 2 sans passer par le centre ?*



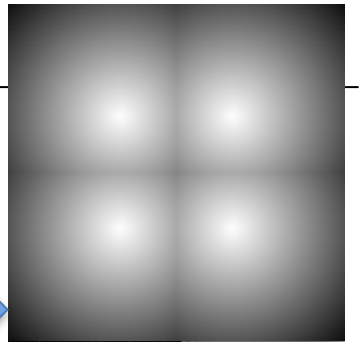
```
line (x0, y0, width-x0, height-y0);
```

1.3. ***(1pt)*** *Comment s'appelle le principe de math utilisé dans les* ***2*** *questions précédentes ?*

Le complémentaire

2. Pixels (11 pts)

Nous partirons du programme ci-dessous qui donne le résultat ci-contre.

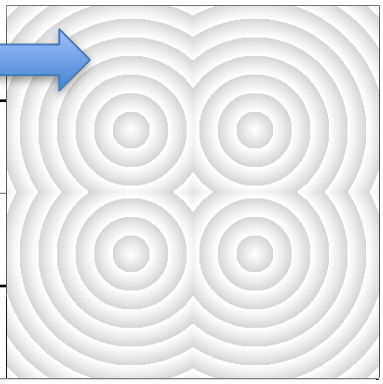


```
1.  void draw(){
2.   loadPixels();
3.     for (int j=0; j<height; j++) {
4.       for (int i=0; i<width; i++) {
5.         float d1 = dist(1*width/3, 1*height/3, i, j) ;
6.         float d2 = dist(2*width/3, 1*height/3, i, j) ;
7.         float d3 = dist(1*width/3, 2*height/3, i, j) ;
8.         float d4 = dist(2*width/3, 2*height/3, i, j) ;
9.         float mini = min(min(d1, d2), min(d3, d4));
10.        pixels[i+j*height] = color(255-mini/(width/2)*255);
11.      }
12.    }
13. }
```

2.1. ***(1pt)*** *Comment s'appelle le principe de math utilisé a la fin de la ligne 10 ?*

Règle de trois (proportionnalité)

2.2. ***(2 pts)*** *plutôt que faire un dégradé de noir et blanc on souhaite que lorsque la distance s'éloigne de 20 pixels la luminosité revienne à 255 et recommence à descendre. Que changer ?*

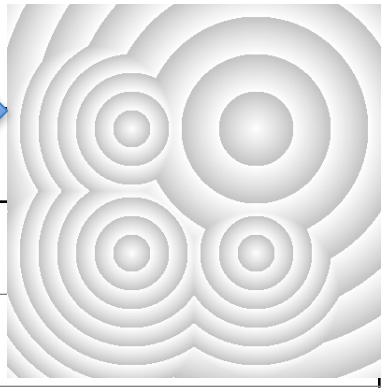


color(255-(mini%20)/(width/2)*255);

On utilise le modulo !

Si vous n'avez pas réussi la question 2.2 vous pouvez quand même faire la suite !

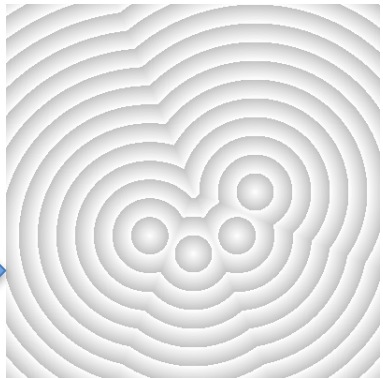
2.3. ***(2 pts)*** *Comment donner plus de « poids » au coin haut-droit ?*



float d2 = 0.5*dist(2*width/3,

1*height/3, i, j) ;

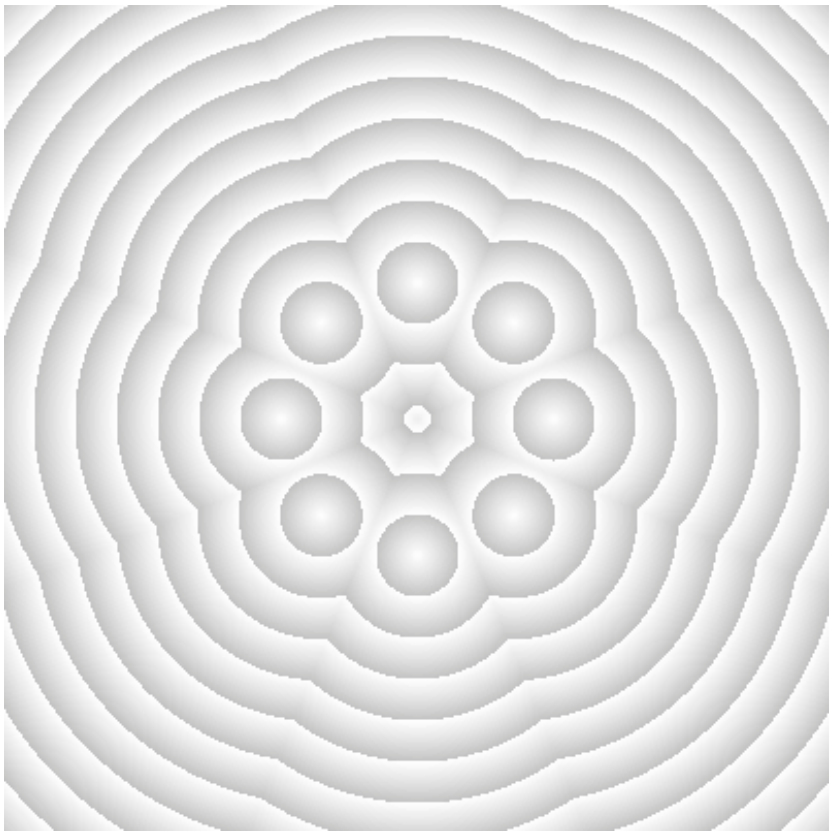
2.4. ***(2 pts)*** *On souhaite maintenant que la position des 4 points ne soient plus codées dans le code par les positions 1/3 et 2/3 mais réparties autour de centre avec 4 angles espacés d'un angle a. Le premier angle pourra être 0 comme ci-contre, ou bien PI/4 pour refaire la même image que ci dessus ? (donner juste la formule pour d1 et d2)*



```
float d1 = dist(width/2+width/6*cos(0*PI/4),
                width/2+width/6*sin(0*PI/4), i, j);
float d2 = dist(width/2+width/6*cos(1*PI/4),
                width/2+width/6*sin(1*PI/4), i, j);
```

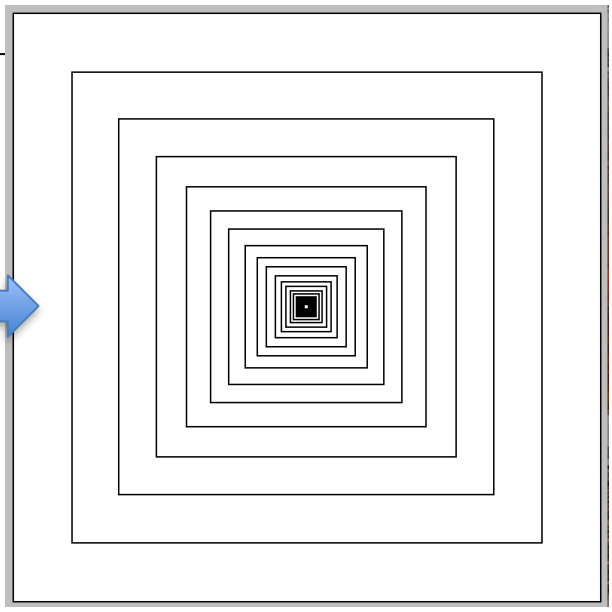
2.5. ***(4 pts)*** *Donner le code qui calcule l'image pour N points (ci dessous N=9 par exemple)*

```
float mini = width;
for (int k=0; k<9; k++){
  float d1 = dist(width/2+width/6*cos(k*PI/4),
                  width/2+width/6*sin(k*PI/4), i, j);
  mini = min(mini, d1);
}
pixels[i+j*height] = color(255-(mini%20)/(width/2)*255);
```



3. Dessin récursif (6 pts)

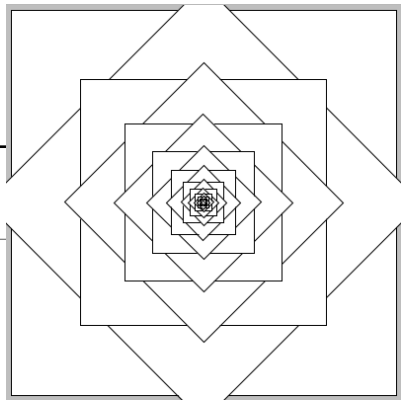
```
1.  void drawR(float R, int i){
2.    rect(0,0, R, R);
3.    // ICI
4.    if (R>4)
5.      drawR(R*0.8, i+1);
6.  }
7.
8.  void draw(){
9.    background(192);
10.   strokeWeight(1);
11.   rectMode(CENTER);
12.   translate(width/2, height/2);
13.   drawR(width-10, 0);
14. }
```



3.1. ***(1pt)*** *A quoi sert la ligne 4?*

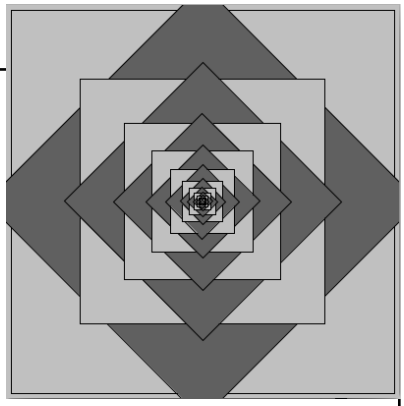
A arrêter la récursivité

3.2. ***(2 pts)*** *Que rajouter à la ligne 3 pour obtenir le résultat ci-contre ?*



rotate(PI/4);

3.3. ***(3 pts)*** *Re-écrivez entièrement le code pour faire un rectangle sur 2 gris foncé et un rectangle sur deux gris clair ? (Vous pourrez écrire sur la copie d'examen si vous n'avez pas assez de place)*



```
void drawR(float R, int i){
  rect(0,0, R, R);
  if (i%2==0) fill(96);
  else fill(192);
  rotate(PI/4);
  if (R>4)
    drawR(R*0.8, i+1);
}

void draw(){
  background(192);
  strokeWeight(1);
  rectMode(CENTER);
  translate(width/2, height/2);
  drawR(width-10, 0);
}
```