

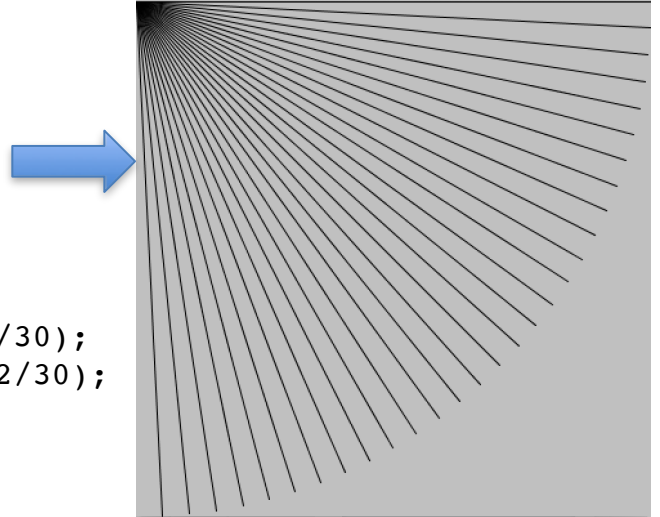
Examen 2022, session rattrapage (2h, barème /20)

1. Dessin (13 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.   float x1 = 0;
9.   float y1 = 0;
10.  for (int i=0; i<30; i++){
11.    float x0 = 0 + width*cos(i*PI/2/30);
12.    float y0 = 0 + height*sin(i*PI/2/30);
13.    line (x1, y1, x0, y0);
14.  }
15. }
```



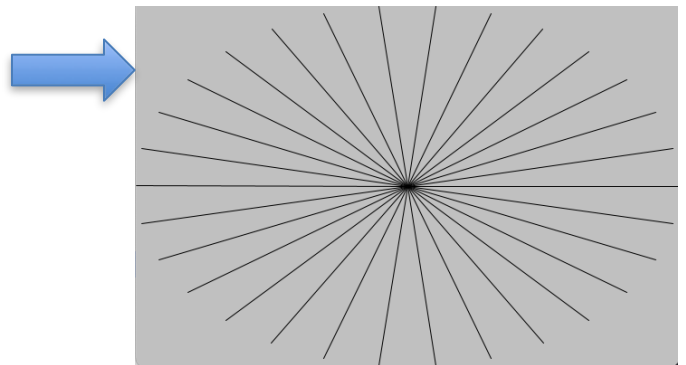
1.1. **(1 pt)** Comment s'appelle le principe de math multiplié i par $2*PI$ et le divise par 30 à l'intérieure des formules de trigonométrie ?

Règle de trois / des proportionalités

1.2. **(2,5 pts)** Ecrivez un programme similaire qui dessine une étoile qui s'adapte à une fenêtre rectangulaire (qui touche les 4 bords)

```

1. void setup(){
2.   size(600, 400);
3. }
4.
5. void draw() {
6.   background(192);
7.   strokeWeight(1);
```



```
float x1 = width/2;
```

```
float y1 = height/2;
```

```
for (int i=0; i<30; i++){
```

```
    float x0 = width/2 + width/2*cos(i*2*PI/30);
```

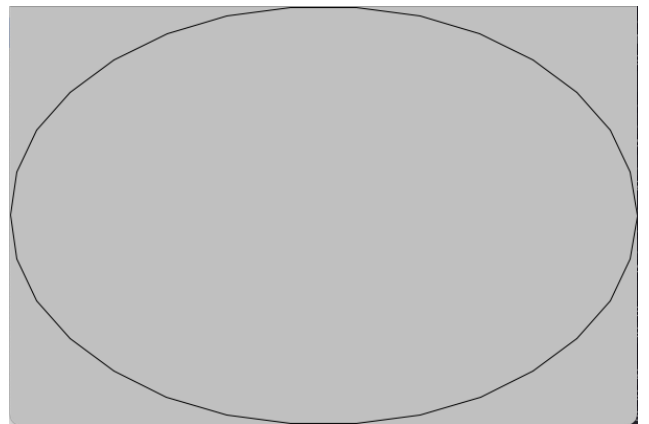
```
    float y0 = height/2 + height/2*sin(i*2*PI/30);
```

```
    line (x1, y1, x0, y0);
```

```
}
```

- 1.3. **(3,5pt)** Ecrivez un programme similaire qui relie les points 2 à 2 pour faire une forme ovale qui s'adapte à une fenêtre rectangulaire (qui touche les 4 bords)

```
1. void setup(){  
2.     size(600, 400);  
3. }  
4.  
5. void draw() {  
6.     background(192);  
7.     strokeWeight(1);
```



```
for (int i=0; i<30; i++){
```

```
    float x0 = width/2 + width/2*cos(i*2*PI/30);
```

```
    float y0 = height/2 + height/2*sin(i*2*PI/30);
```

```
    float x1 = width/2 + width/2*cos((i+1)*2*PI/30);
```

```
    float y1 = height/2 + height/2*sin((i+1)*2*PI/30);
```

```
    line (x1, y1, x0, y0);
```

```
}
```

- 1.4. **(6pt)** Ecrivez un nouveau programme qui re-utilise ce que vous avez déjà fait en écrivant quelle lignes vous re-utilisez et où (= avant/apres/au milieu) pour faire une toile d'araignée qui s'adapte à la forme rectangulaire de l'écran

```
1. void setup(){  
2.     size(400, 400);  
3. }  
4.
```

```

5. void draw() {
6.     background(192);
7.     strokeWeight(1);

```

// On commence par re-utiliser le code de la 1.1

```
for (float j=1; j<=10; j++){
```

```
    for (int i=0; i<30; i++){
```

```
        x0 = width/2 + width/2*j/10*cos(i*2*PI/30);
```

```
        y0 = height/2 + height/2*j/10*sin(i*2*PI/30);
```

```
        x1 = width/2 + width/2*j/10*cos((i+1)*2*PI/30);
```

```
        y1 = height/2 + height/2*j/10*sin((i+1)*2*PI/30);
```

```
        line (x1, y1, x0, y0);
```

```
    }
```

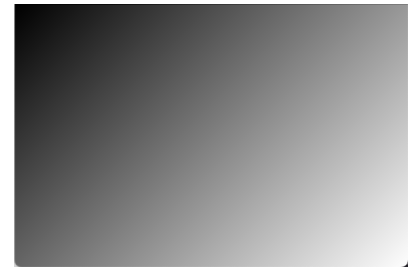
2. Pixels (7 pts)

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

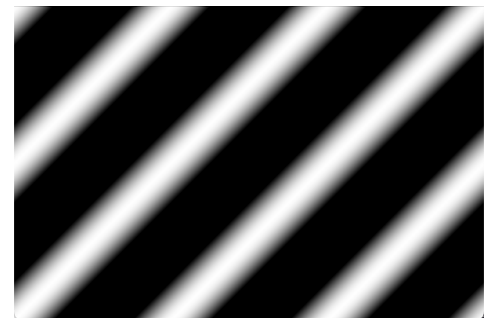
```

1. void draw(){
2.     loadPixels();
3.     for (int j=0; j<height; j++) {
4.         for (int i=0; i<width; i++) {
5.             float truc = (i+j)*1.0/(width+height);
6.             pixels[i+j*width] = color(truc*255);
7.         }
8.     }
9.     updatePixels();
10.

```



2.1. (2 pts) plutôt que faire un dégradé de noir et blanc sur toute la diagonale on souhaite utiliser une fonction de trigonométrie sur la variable "truc" pour faire un motif de 5 tubes (NB la fonction color(...) sur un chiffre négatif rend la couleur noire). Ne re-écrivez que les lignes 5 et/ou 6 du programme



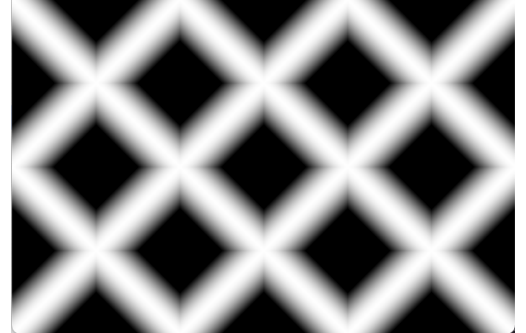
pixels[i+j*width] =

color(cos(truc*PI*10)*255);

On utilise le cosinus

2.2. **(5 pts)** En utilisant la fonction `maximum` croisez des tubes sur les 2 diagonales. Ne re-écrivez que les lignes 5 et/ou 6 du programme

PS: si vous ne savez pas le faire, vous pouvez faire les diagonales dans l'autre sens pour 2pts



<code>float truc = (i+j)*1.0/(width+height);</code>
<code>float machin = (width-i+j)*1.0/(width+height);</code>
<code>pixels[i+j*width] = color(max (cos(truc*PI*10)*255,</code>
<code>cos(machin*PI*10)*255));</code>