

Modification et reprise d'un code issu du Livre Générative Design par Abram Libr

```
// * KEYS
// * 1-3      : create a new collage (layer specific)
// * s       : save png

import java.util.Calendar;

PImage[] images; //classe images
String[] imageNames; //affichage du nom des images et décompte
int imageCount;    //variable permettant de compter le nombre d'images
PImage photo1;    //variable pour afficher le titre "Exposition Exposée"
PImage photo2;    //variable pour affiché la date
int x;
int y;
int a;
int b;

CollageItem[] layer1Items, layer2Items, layer3Items; //annonce des différents calques sur lesquelles s'affiche-
ront les images

void setup() {

    // Création de la page de l'affiche, arrière plan blanc, titre, date
    size(2480, 3508);
    imageMode(CENTER); //placement des images selon leur centre
    background(255);

    photo1 = loadImage("titre.jpg"); // chargement du titre
    photo2 = loadImage("date.jpg"); // chargement de la date

    // Charger les images dans une classe

    File dir = new File(sketchPath(""), "../P_4_2_1_footage");

    if (dir.isDirectory()) {
        String[] contents = dir.list();
        images = new PImage[contents.length];
        imageNames = new String[contents.length];
        for (int i = 0 ; i < contents.length; i++) { //chargements des différentes images unes à unes

            if (contents[i].charAt(0) == '.') continue;
            else if (contents[i].toLowerCase().endsWith(".jpg")) {
                File childFile = new File(dir, contents[i]);
                images[imageCount] = loadImage(childFile.getPath()); //Le code va chercher les images "jpg" dans le
                dossier
                imageNames[imageCount] = childFile.getName();
                println(imageCount+" "+contents[i]+" "+childFile.getPath()); //Le code va afficher le nom des différents
```

images dans la console

```
    imageCount++;  
  }  
}  
}
```

// Placement aléatoire des images des différents calques dans l'affiche

```
layer1Items = generateCollageItems("layer1", 100, 0,height/2, TWO_PI,height, 0.1,0.5, 0,0);  
layer2Items = generateCollageItems("layer2", 150, 0,height/2, TWO_PI,height, 0.1,0.3, -PI/6,PI/6);  
layer3Items = generateCollageItems("layer3", 110, 0,height/2, TWO_PI,height, 0.1,0.2, 0,0);
```

// Faire afficher les images des différents calques (voir dans le void draw)

```
drawCollageItems(layer1Items);  
drawCollageItems(layer2Items);  
drawCollageItems(layer3Items);  
}
```

```
void draw() {
```

// Titre et date

```
image(photo1,1240,400,2480, 820);  
image(photo2,1240,3300,2480, 390);
```

```
}
```

// Interaction et affichage des images avec les touche du clavier

```
void keyPressed() {  
  if (key == 's' || key == 'S') saveFrame(timestamp()+"_####.jpg"); //sauvegarde de l'affiche  
  
  if (key == '1') layer1Items = generateCollageItems("layer1", (int)random(50,200), 0,height/2,PI*5,height,  
0.1,0.5, 0,0); //changement de place et d'images du calque 1  
  if (key == '2') layer2Items = generateCollageItems("layer2", (int)random(25,300), 0,height*0.15,PI*5,150,  
0.1,random(0.3,0.8), -PI/6,PI/6); //changement de place et d'images du calque 2  
  if (key == '3') layer3Items = generateCollageItems("layer3", (int)random(50,300),  
0,height*0.66,PI*5,height*0.66, 0.1,random(0.2,0.5), -0.05,0.05); //changement de place et d'images du calque  
3  
}
```

// Affichage des images

```
background(255);  
drawCollageItems(layer1Items);  
drawCollageItems(layer2Items);  
drawCollageItems(layer3Items);  
}
```

// Générer un collage et gérer la place des images dans l'affiche

```
class CollageItem {  
  float a = 0, l = 0;  
  float scaling = 1;  
  int indexToImage = -1;  
}
```

```
// Assistance de chargement des attributs de chaque images
```

```
CollageItem[] generateCollageItems(String thePrefix, int theCount,  
float theAngle, float theLength, float theRangeA, float theRangeL,  
float theScaleStart, float theScaleEnd, float theRotationStart, float theRotationEnd) {  
    //collecte des images dans chaque calques  
    int[] indexes = new int[0];  
    for (int i = 0 ; i < imageNames.length; i++) {  
        if (imageNames[i] != null) {  
            if (imageNames[i].startsWith(thePrefix)) {  
                indexes = append(indexes, i);  
            }  
        }  
    }  
}
```

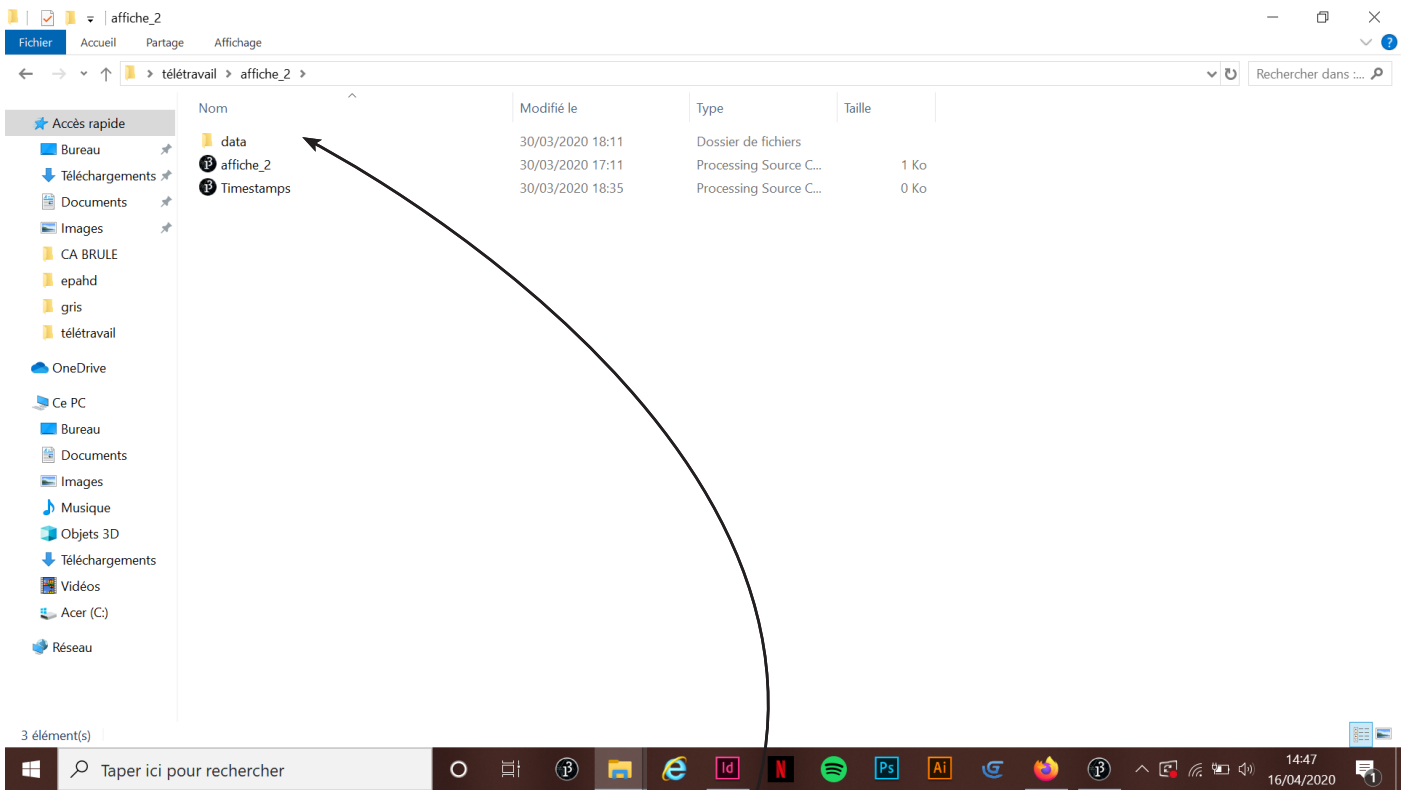
```
// Affichage de nouvelles images
```

```
CollageItem[] items = new CollageItem[theCount];  
for (int i = 0 ; i < items.length; i++) {  
    items[i] = new CollageItem();  
    items[i].indexToImage = indexes[i%indexes.length];  
    items[i].a = theAngle + random(-theRangeA/2,theRangeA/2);  
    items[i].l = theLength + random(-theRangeL/2,theRangeL/2);  
    items[i].scaling = random(theScaleStart,theScaleEnd);  
  
}  
return items;  
}
```

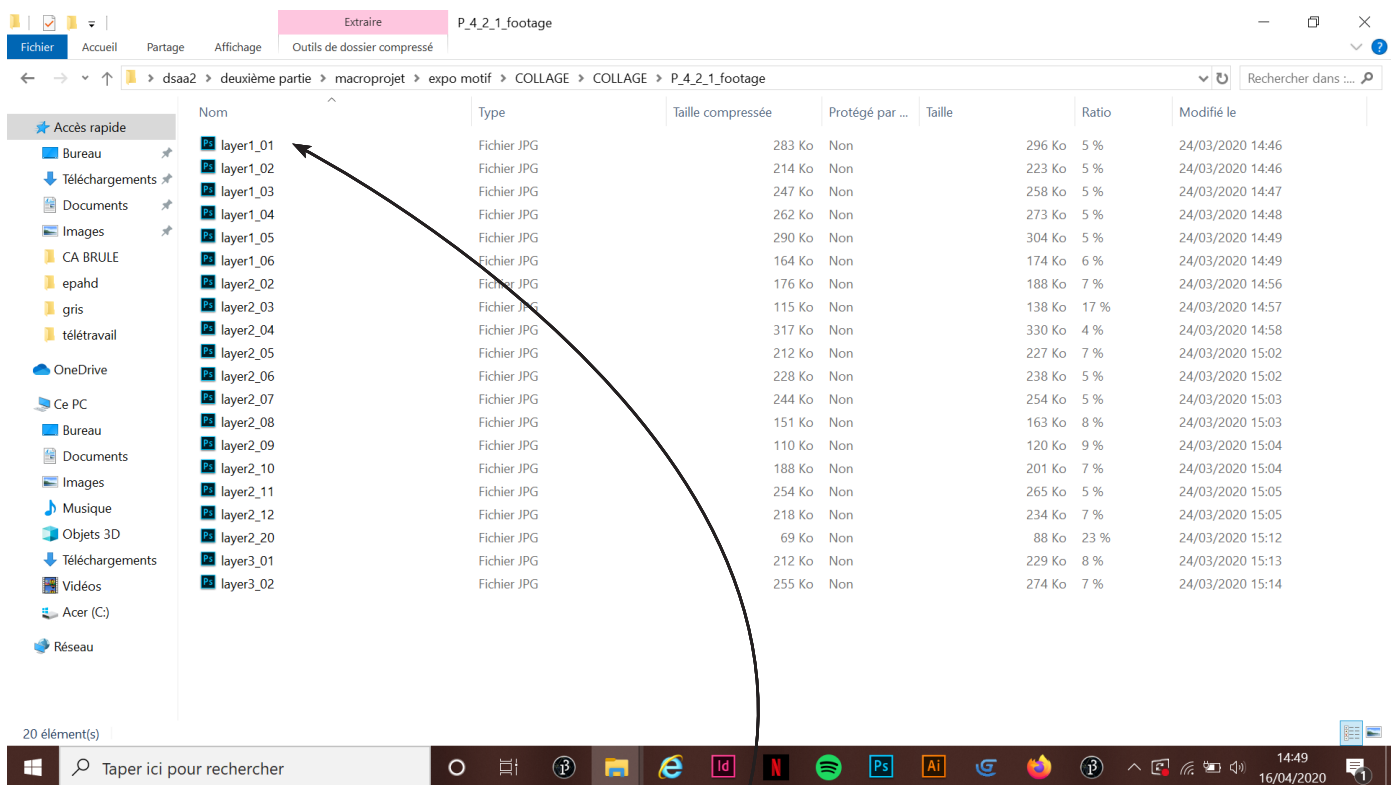
```
void drawCollageItems(CollageItem[] theItems) {  
    for (int i = 0 ; i < theItems.length; i++) {  
        pushMatrix();  
        float newX = width/2 + cos(theItems[i].a) * theItems[i].l;  
        float newY = height/2 + sin(theItems[i].a) * theItems[i].l;  
        translate(newX, newY);  
        scale(theItems[i].scaling);  
        image(images[theItems[i].indexToImage], 0,0);  
        popMatrix();  
    }  
}
```

```
// timestamp, il permet de généré un png de l'image quand on appui sur "S"
```

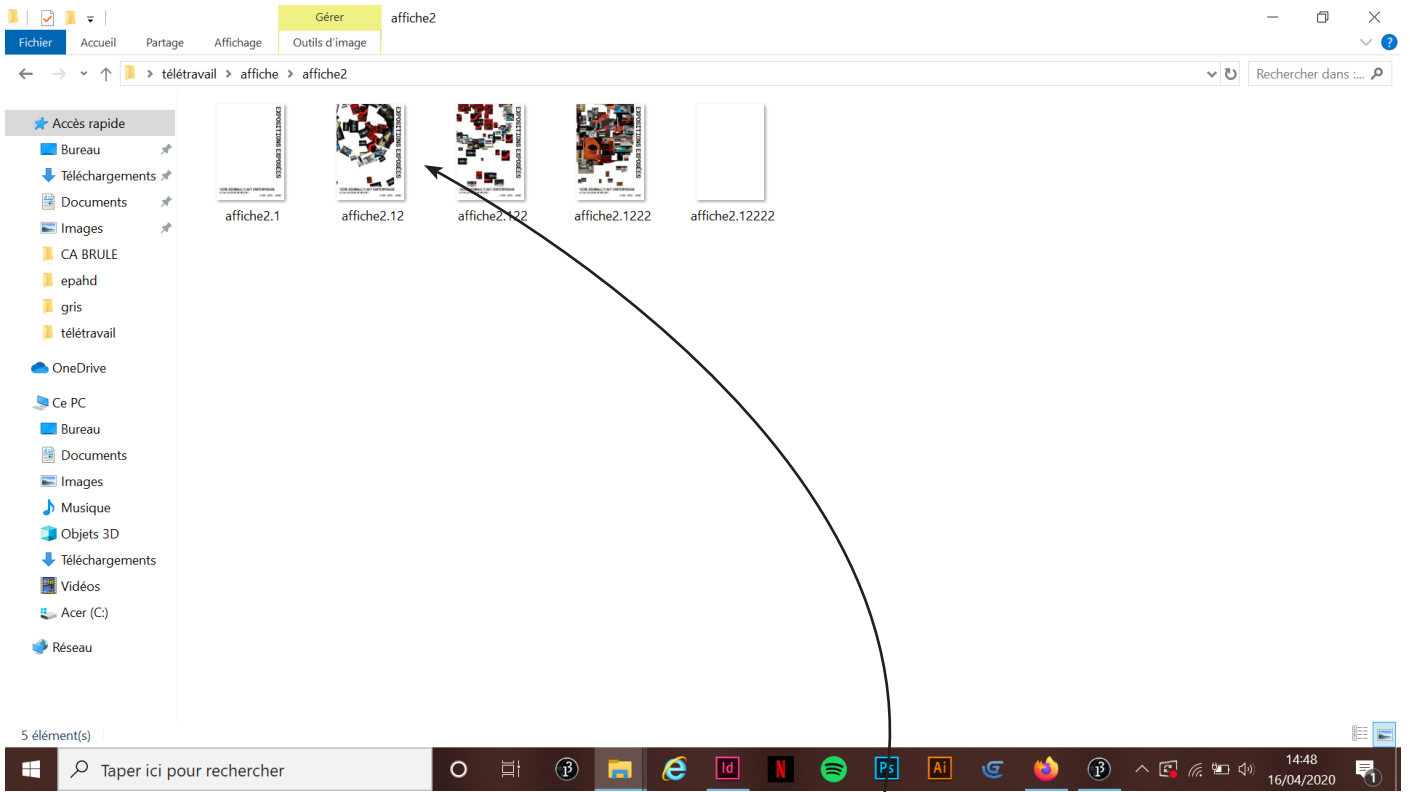
```
String timestamp() {  
    Calendar now = Calendar.getInstance();  
    return String.format("%1$tY%1$tm%1$td_%1$tH%1$tM%1$tS", now);  
}
```



Glisser déposer l'image dans le fichier «data»



renommer l'image «layer1_ «...»»
Les autres calques apparaissent en dessous,
on pourra y trouver les images
plus anciennes.



Affiche générée dans un fichier spécifique (possibilité de l'imprimer).