

Une carte de vœux avec Processing

Objectif

Le programme permet de construire une grille de carrés de 4 par 6 dont la couleur est fixée aléatoirement. Dans chaque carré une lettre est dessinée dans la couleur complémentaire.



A chaque exécution du programme vous obtiendrez une carte différente !

Choix de la police de caractères

Pour choisir une police de votre système il est nécessaire de recourir à un petit programme qui liste les polices.

```
sketch_181126a
1 // Liste des polices
2 void setup() {
3   String[] fontList = PFont.list();
4   println(fontList);
5 }

[0] "1454 Gutenberg Bibel"
[1] "Agency FB"
[2] "Agency FB Gras"
[3] "Algerian"
[4] "Arial"
[5] "Arial Black Normal"
[6] "Arial Bold"
[7] "Arial Bold Italic"
[8] "Arial Gras"
[9] "Arial Gras Italique"
[10] "Arial Italic"
[11] "Arial Italique"
[12] "Arial Narrow"
[13] "Arial Narrow Gras"
[14] "Arial Narrow Gras Italique"
[15] "Arial Narrow Italic"
[16] "Arial Rounded MT Bold"
[17] "Bahnschrift"
[18] "Bahnschrift Bold"
[19] "Bahnschrift Bold Condensed"
```

Code du programme :

```
// Liste des polices
void setup() {
String[] fontList = PFont.list();
println(fontList);
}
```

Choisissez la police qui vous convient dans la liste, je vous suggère à la ligne 5 :

```
[5] "Arial Black Normal"
```

Le nom à retenir est celui qui est entre les apostrophes, il doit être écrit à l'identique : Arial
Black Normal

Carte de vœux

Déclaration des variables

Définition des carrés :

```
int vert = 4;          // nombre de carrés à la verticale
int hori = 6;         // nombre de carrés à l'horizontale
int cote = 200;       // taille des carrés en pixels
int demi = cote/2;    // taille de la moitié du carré
size(1200, 800);     // taille de la fenêtre (cote*hori, cote*vert)
rectMode(CENTER);    // point d'origine des carrés au centre
noStroke();          // les carrés n'ont pas de cadre
```

Définition de la police :

```
PFont myFont;         // initialisation de la classe "Font"
int tpol = 170;       // taille de la police
int dpol = cote/4;    // décalage des caractères au centre des carrés
myFont = createFont("Arial Black Normal", tpol); // initialisation police
String cars="Joyeux Noël Bonne Année"; // texte de la carte
String car;          // lecture du texte caractère par caractère
textFont(myFont);    // activation de la police
textAlign(CENTER, CENTER); // point d'origine des caractères au centre
smooth();            // antialiasing sur les caractères
```

Autres variables :

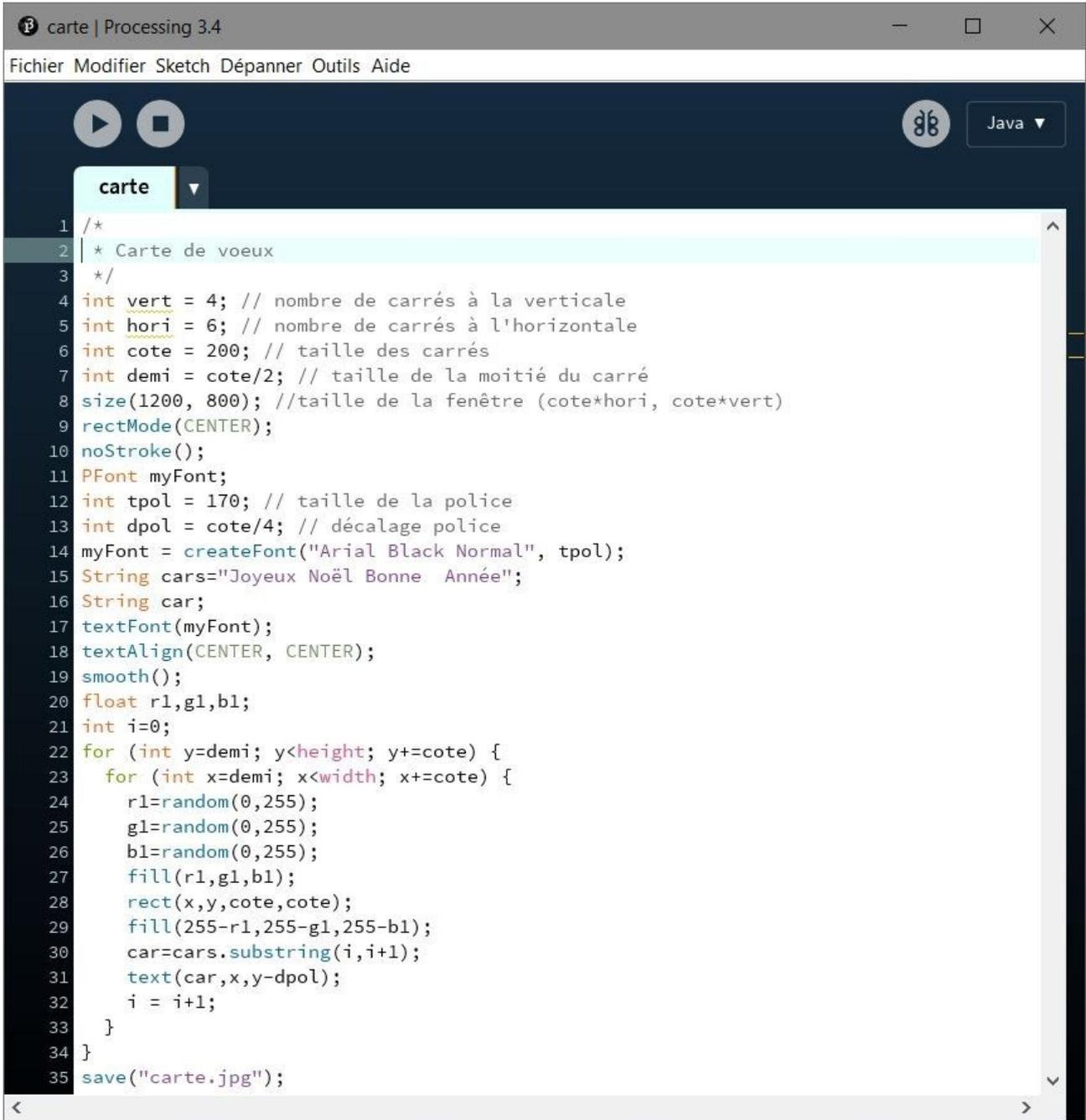
```
float r1,g1,b1; // couleurs au format RGB (voir w3schools)
int i=0;        // index pour le parcours du texte
```

Affichage de la grille et du texte

```
for (int y=demi; y<height; y+=cote) {
  for (int x=demi; x<width; x+=cote) {
    r1=random(0,255); // valeur de la composante rouge
    g1=random(0,255); // valeur de la composante verte
    b1=random(0,255); // valeur de la composante bleue
    fill(r1,g1,b1);   // couleur du carré
    rect(x,y,cote,cote); // dessin du carré
    fill(255-r1,255-g1,255-b1); // couleur du texte (complémentaire)
    car = cars.substring(i,i+1); // extraction du caractère d'index i
    text(car,x,y-dpol); // dessin du caractère
    i = i+1;           // incrémentation de l'index
  }
}
```

Les carrés sont affichés ligne par ligne. Les couleurs sont définies aléatoirement à l'aide de la fonction `random()`. La couleur des caractères est inversée par rapport à la couleur des carrés.

Code complet



```
1 /*
2  * Carte de voeux
3  */
4 int vert = 4; // nombre de carrés à la verticale
5 int hori = 6; // nombre de carrés à l'horizontale
6 int cote = 200; // taille des carrés
7 int demi = cote/2; // taille de la moitié du carré
8 size(1200, 800); //taille de la fenêtre (cote*hori, cote*vert)
9 rectMode(CENTER);
10 noStroke();
11 PFont myFont;
12 int tpol = 170; // taille de la police
13 int dpol = cote/4; // décalage police
14 myFont = createFont("Arial Black Normal", tpol);
15 String cars="Joyeux Noël Bonne Année";
16 String car;
17 textFont(myFont);
18 textAlign(CENTER, CENTER);
19 smooth();
20 float r1,g1,b1;
21 int i=0;
22 for (int y=demi; y<height; y+=cote) {
23   for (int x=demi; x<width; x+=cote) {
24     r1=random(0,255);
25     g1=random(0,255);
26     b1=random(0,255);
27     fill(r1,g1,b1);
28     rect(x,y,cote,cote);
29     fill(255-r1,255-g1,255-b1);
30     car=cars.substring(i,i+1);
31     text(car,x,y-dpol);
32     i = i+1;
33   }
34 }
35 save("carte.jpg");
```

A vous de compléter ce programme, et quand vous aurez fini de le personnaliser n'oubliez pas d'ajouter à la fin la ligne :

```
save("carte.jpg");
```

Cette commande permet d'enregistrer l'image dans le répertoire du projet sous le nom "carte" au format JPG.