

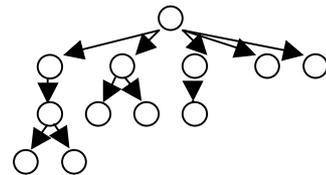
ED 12 - Treemaps

P. Cubaud <cubaud @ cnam.fr>

Les Treemaps ont été inventés par B. Schneiderman (Université du Maryland) en 1991. Il s'agit d'une technique de visualisation de grandes arborescences (systèmes de fichiers, taxonomies, etc.). Depuis ce premier travail, de nombreux perfectionnements ont été apportés et cette méthode se diffuse progressivement dans l'industrie (du Web, en particulier).

Exercice 1 : Premier exemple

Produire le treemap de l'arbre suivant :



Exercice 2 : Génération aléatoire d'arbre

Quelles structures de données sont utiles pour générer aléatoirement un arbre ?

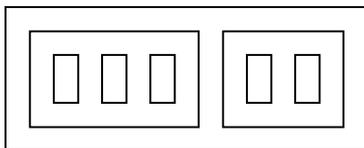
Quelles sont les informations à produire pour le dessin du treemap ?

Donner un algorithme qui produit ces informations pour tous les sommets de l'arbre

Exercice 3 : Algorithme de subdivision en largeur

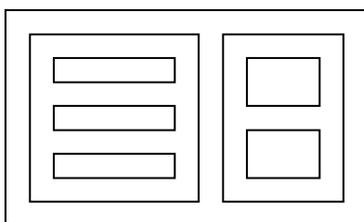
On suppose résolu l'ex. précédent. Donner un premier programme Processing de dessin du treemap, où seule la largeur du rectangle du sommet père est partagée pour les sommets fils.

On fixera un espacement constant entre les rectangles :



Exercice 4 : Subdivision en largeur et hauteur alternées ("slice and dice")

Modifier le programme précédent pour que la subdivision alterne partage de la largeur puis partage de la hauteur. C'est l'algorithme de départ de B. Schneiderman.



Exercice 5 : Coloriage du treemap

En général, les couleurs des cases du treemap dépendent d'informations associées aux sommets de l'arbre. Proposer une technique générique de coloriage.

corrigé exercice 2

```
int NSOM = 10;
int[] pere = new int[NSOM];
int[] poids = new int[NSOM];
int[][] fils = new int[NSOM][NSOM];

// generation de l'arbre aleatoire
void generation(){
  for (int i=0; i<NSOM; i++){
    pere[i] = int(random(0,i-1));
    poids[i] = 0;
    for (int j=0; j<NSOM; j++){
      fils[i][j] = -1;
    }
  }
  pere[0] = -1;

  // recherche des fils
  for (int i=0; i<NSOM; i++){
    int n=0;
    for (int j=1; j<NSOM; j++){
      if (pere[j] == i) {fils[i][n]=j;n++;}
    }
  }
}

// calcul des poids
int calpoids(int i){
  int pds;
  if (fils[i][0] == -1) {poids[i] = 1; return 1;} // feuille
  else {
    pds = 0;
    for (int j=0; j<NSOM; j++){
      if (fils[i][j] >= 0) pds += calpoids(fils[i][j]);
    }
    poids[i] = pds;
    return pds;
  }
}
```

corrigé exercice 3

```
void setup(){
  generation();
  int r = calpoids(0);
  size(400,300);
  noFill();stroke(0);background(255);
  dessineFils(0,0,0,float(width),float(height));
}
```

```
void draw(){}
```

```
void dessineFils(int n, int x, int y, float l, float h){  
    rect(x,y,l,h);  
    float a = l/poids[n];  
    int i = 0;  
    while (fils[n][i] > 0){  
        float lfils = a*poids[fils[n][i]];  
        float hfils = h;  
        dessineFils(fils[n][i], x+epsil,y+epsil,lfils-2*epsil,hfils-2*epsil);  
        i++;  
        x += lfils; // y inchangé  
    }  
}
```

// + ici les fonctions de l'exercice 1

corrigé exercice 4

```
void dessineFils(int n, boolean enlargeur,int x, int y, float l, float h){  
    rect(x,y,l,h);  
    if (enlargeur) {  
        float a = l/poids[n];  
        int i = 0;  
        while (fils[n][i] > 0){  
            float lfils = a*poids[fils[n][i]];  
            float hfils = h;  
            dessineFils(fils[n][i], false, x+epsil,y+epsil,lfils-2*epsil,hfils-2*epsil);  
            i++;  
            x += lfils; // y inchangé  
        }  
    }  
    else {  
        float a = h/poids[n];  
        int i = 0;  
        while (fils[n][i] > 0){  
            float hfils = a*poids[fils[n][i]];  
            float lfils = l;  
            dessineFils(fils[n][i], true, x+epsil,y+epsil,lfils-2*epsil,hfils-2*epsil);  
            i++;  
            y += hfils; // x inchangé  
        }  
    }  
}
```

corrigé exercice 5

au début de dessineFils

```

void dessineFils(int n, boolean enlagueur,int x, int y, float l, float h){
  if (pere[n] == 0) {teinte = teinte + int(deltateinte); fill(teinte,100,100); println(n+ " :
"+teinte);}
  else if (pere[n] > 0) fill(teinte,100*(NSOM-
n)/NSOM,100*n/NSOM);//fill(teinte,100*poids[n]/poids[pere[n]],100*(NSOM-n)/NSOM);
  else fill(0,0,0);
  rect(x,y,l,h);

```

+ nouvelle version de setup() :

```

void draw(){
  stroke(255);
  background(255);
  generation();
  int r = calpoids(0);
  // calcul couleurs de depart
  int i = 0;
  while (fils[0][i] > 0){i++;}
  deltateinte = 360.0/i;
  teinte = 0;
  println(deltateinte);
  dessineFils(0,true,0,0,float(width),float(height));
}

```