

Tris, matrices, animations et mini-jeux

Valeur d'accueil et de reconversion en informatique (VARI1)

Daniel Porumbel (dp.cnam@gmail.com)

<http://cedric.cnam.fr/~porumbed/vari1/>

- 1 Un exemple de tri de tableau
- 2 Un exemple de matrice (tableau de tableaux)
- 3 Mini-jeu casse-brique avec une balle

```
int [] tab;  
void mettreMinAuPremierIndice (int premierIndice) {  
    int indiceMin = premierIndice;  
  
    for (int i=premierIndice+1; i<tab.length; i++)  
        if (tab[i]<tab[indiceMin])  
            indiceMin = i;  
  
    int tmp = tab[premierIndice];  
    tab[premierIndice] = tab[indiceMin];  
    tab[indiceMin] = tmp;  
}
```

- 1 Quel est l'action de cette méthode ? Donner un exemple
- 2 Comment l'utiliser pour faire un tri ?

Trier avec une boucle explicite 1

```
int [] tab;
void mettreMinAuPremierIndice (...) {
    //mettre le code du slide précédent
}
void setup () {
    tab = new int [20];
    for (int i=0; i<tab.length; i++)
        tab[i] = (int)random(50);
    //Affichage tableau de départ
    println (java.util.Arrays.toString (tab));
    //Tri:
    for (int i=0; i<tab.length; i++)
        mettreMinAuPremierIndice (i);
    //Affichage tableau trié
    println (java.util.Arrays.toString (tab));
}
```

Trier avec une boucle `draw()` implicite 1

```
void mettreMinAuPremierIndice (...) {  
} ...  
void setup() {  
    tab = new int[20];  
    for(int i=0; i<tab.length; i++)  
        tab[i] = (int)random(50);  
    frameRate(1); //1 draw par seconde  
}  
int i=0;  
void draw() {  
    mettreMinAuPremierIndice(i);  
    if(i<tab.length-1)  
        i++;  
    //afficher le tableau  
    println(java.util.Arrays.toString(tab));  
}
```

Trier avec une boucle `draw()` implicite 2

```
void histo() {  
    for (int i=0; i<tab.length; i++){  
        fill(255,255,255);  
        rect(0,20*i,tab[i],20);  
        fill(255,0,0);  
        text(tab[i],tab[i]/2-15,20*i+10);  
    }  
}
```

- 1 Qu'affiche cette fonction ?
- 2 Peut-on remplacer `println(...)` avec un histogramme ?

Trier avec une boucle `draw()` implicite 2

```
void histo() {  
    for (int i=0; i<tab.length; i++){  
        fill(255,255,255);  
        rect(0,20*i, tab[i],20);  
        fill(255,0,0);  
        text(tab[i], tab[i]/2-15,20*i+10);  
    }  
}
```

- 1 Qu'affiche cette fonction ?
- 2 Peut-on remplacer `println(...)` avec un histogramme ?

- 1 Un exemple de tri de tableau
- 2 Un exemple de matrice (tableau de tableaux)**
- 3 Mini-jeu casse-brique avec une balle

1 Le code ci-après initialise une matrice, c.-à.-d. un tableau de tableaux

```
int [][] mat = new int [10][10]; // matrice 10 X 10
for (int i=0; i<10; i++)
    for (int j=0; j<10; j++)
        mat[i][j] = i*j;
```

2 Voici une méthode qui affiche une matrice dans la console

- `m.length` est le nombre de lignes
- `java.util.Arrays.toString(m[i])` affiche la ligne *i*

```
void afficherMatrice(int [][] m){
    for(int i=0; i<m.length; i++)
        println(java.util.Arrays.toString(m[i]));
}
```

1 Le code ci-après initialise une matrice, c.-à.-d. un tableau de tableaux

```
int [][] mat = new int [10][10]; // matrice 10 X 10
for (int i=0; i<10; i++)
    for (int j=0; j<10; j++)
        mat[i][j] = i*j;
```

2 Voici une méthode qui affiche une matrice dans la console

- `m.length` est le nombre de lignes
- `java.util.Arrays.toString(m[i])` affiche la ligne *i*

```
void afficherMatrice (int [][] m) {
    for (int i=0; i<m.length; i++)
        println (java.util.Arrays.toString(m[i]));
}
```

3 Quel est le résultat de ce programme ?

```
void drawMatrice(int [][] m){
    textSize(20);
    for(int i=0; i<m.length; i++) //9
        for(int j=0; j<m[i].length; j++)
            text(m[i][j], j*40, i*40+40);
}

void setup(){
    int [][] mat = new int [9][10]; //9 X 10
    for(int i=0; i<9; i++)
        for(int j=0; j<10; j++)
            mat[i][j] = i*j;
    size(400,400);
    fill(0,0,255);
    drawMatrice(mat);
}
```

- 1 Un exemple de tri de tableau
- 2 Un exemple de matrice (tableau de tableaux)
- 3 Mini-jeu casse-brique avec une balle**

Quel est le résultat du code ?

```
PImage laBalle ;
void setup() {
    laBalle = loadImage("balle.jpeg");
    size(1000,600);
}
int x = 0; int y = 400;
int deltaX = 5; int deltaY = 5;
void draw() {
    background(200,200,200);
    x = x + deltaX;
    y = y + deltaY;
    if (y<0)
        deltaY=5; //se déplacer en bas
    if (x<0)
        deltaX=5; //se déplacer à droite
    if (x>width-50)
        deltaX=-5; //se déplacer en haut
    if (y>height-50)
        deltaY=-5; //se déplac. à gauche
    image(laBalle , x, y);
}
```

1 Ajouter une palette de longueur 200 centrée à mouseX

- si la balle tombe sur la palette, la balle est renvoyé en haut
- si la balle touche le bord en bas, on affiche « perdu » et on appelle noLoop() (stop jeu)

2 Ajouter des briques en haut

Quel est le résultat du code ?

```
PImage laBalle;  
void setup() {  
    laBalle = loadImage("balle.jpeg");  
    size(1000,600);  
}  
int x = 0; int y = 400;  
int deltaX = 5; int deltaY = 5;  
void draw() {  
    background(200,200,200);  
    x = x + deltaX;  
    y = y + deltaY;  
    if (y<0)  
        deltaY=5; //se déplacer en bas  
    if (x<0)  
        deltaX=5; //se déplacer à droite  
    if (x>width-50)  
        deltaX=-5; //se déplacer en haut  
    if (y>height-50)  
        deltaY=-5; //se déplac. à gauche  
    image(laBalle, x, y);  
}
```

1 Ajouter une palette de longueur 200 centrée à mouseX

- si la balle tombe sur la palette, la balle est renvoyé en haut
- si la balle touche le bord en bas, on affiche « perdu » et on appelle `noLoop()` (stop jeu)

2 Ajouter des briques en haut

Quel est le résultat du code ?

```
PImage laBalle;  
void setup() {  
    laBalle = loadImage("balle.jpeg");  
    size(1000,600);  
}  
int x = 0; int y = 400;  
int deltaX = 5; int deltaY = 5;  
void draw() {  
    background(200,200,200);  
    x = x + deltaX;  
    y = y + deltaY;  
    if (y<0)  
        deltaY=5; //se déplacer en bas  
    if (x<0)  
        deltaX=5; //se déplacer à droite  
    if (x>width-50)  
        deltaX=-5; //se déplacer en haut  
    if (y>height-50)  
        deltaY=-5; //se déplac. à gauche  
    image(laBalle, x, y);  
}
```

1 Ajouter une palette de longueur 200 centrée à mouseX

- si la balle tombe sur la palette, la balle est renvoyé en haut
- si la balle touche le bord en bas, on affiche « perdu » et on appelle `noLoop()` (stop jeu)

2 Ajouter des briques en haut