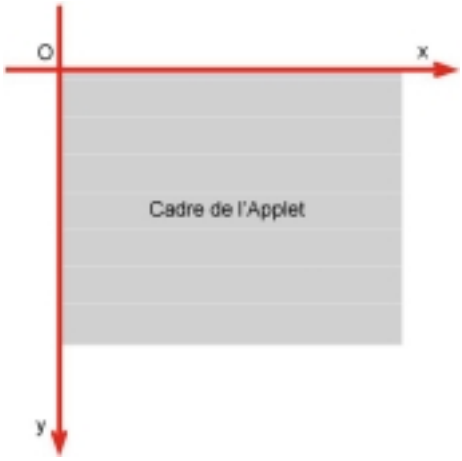
Applets (Classe Graphics) : applet *Floride*

Introduction

Ce chapitre présente les rudiments graphiques disponibles pour une applet. Nous verrons comment dessiner des lignes, rectangles, ovales, arcs de cercle, chaînes de texte…
Avant de commencer, il faut savoir qu'une applet est en gros une application exécutée par l'intermédiaire d'un navigateur web, qui définit automatiquement un cadre de travail. Les dimensions de ce cadre sont définis par le code HTML accompagnant l'applet.
La plupart de ces procédés appartiennent à Java 1.1. Dans Java 2, on a une classe Graphics2D beaucoup plus performante, mais dans la mesure où elle n'est pas utilisable pour les applets destinées à IE5 ou Netscape 4.7, la classe Graphics vous sera bien utile pour des graphiques simples.

Système de coordonnées

On place des éléments dans le cadre d'une applet en définissant leur position par des coordonnées. Le repère a pour origine le point le plus en haut et le plus à gauche du cadre.



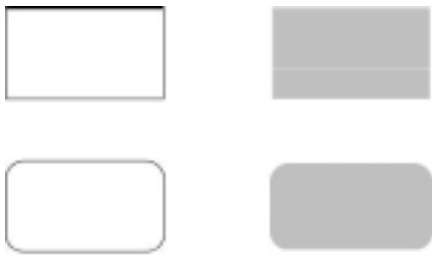
Segments de droites

Pour tracer le segment de droite [AB], on utilise la commande *DrawLine(xA,yA,xB,yB)*.

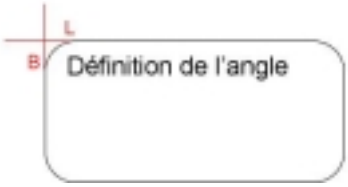
Rectangles

Il existe 4 types de rectangles : rectangle aux angles droits vide et plein, rectangle aux angles arrondis vide ou plein.

- drawRect(x,y,L,h) ; //
- fillRect(x,y,L,h) ;
- drawRoundRect(x,y,L,h,β,λ) ;
- fillRoundRect(x,y,L,h, β,λ);



(x,y) sont les coordonnées de l'angle supérieur gauche du rectangle.
L et h sont la largeur et la hauteur du rectangle.
β et λ sont les valeurs de largeur et de hauteur de l'angle.



Polygones

Les polygones sont des ensembles de segments de droites joints. Il y a 2 types de polygones : les vides et les pleins (comme les rectangles). On peut les construire de 2 manières : soit en définissant manuellement tous les segments qui définissent le polygone soit en créant une matrice de coordonnées x et une autre de y.

- polygon(int[ ], int[ ], int) ;

Le 3° entier représente le nombre de points du polygone. Un petit exemple (pas tout à fait fonctionnel) :

- int x[ ] = {10, 20, 30, 40, 50} ;
- int y[ ] = {50, 60, 70, 80, 90} ;
- int points = x.length ;
- Polygon poly = new Polygon(x, y, points);
- *.drawPolygon(poly);

L'astérisque peut changer de valeurs, on trouve souvent *screen* ou *g*. Nous verrons cela plus tard.

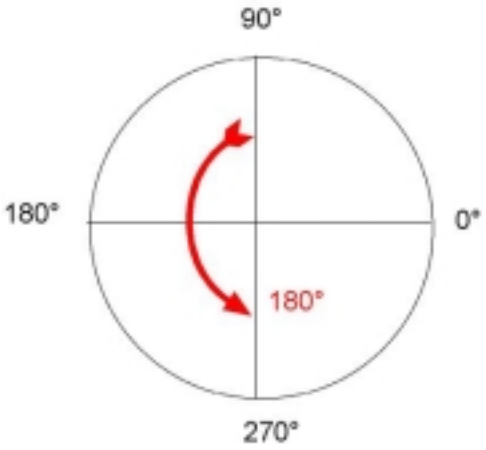
Ovales

Soit (x,y) les coordonnées du point supérieur gauche,

- drawOval(x,x,largeur,hauteur) ;
- fillOval ;

Arcs de cercle

Là, il faut un schéma…



- drawArc(x,y,largeur,hauteur,angle,nombreDeDegrès) ;
- fillArc() ;

(x,y) les coordonnées du point haut gauche
angle = l'angle à partir duquel l'arc commence (ici, 90°)
nombreDeDegrès = nombre de degrés que parcourt l'arc (ici, 180°)

Copier/Couper

Ce n'est pas vraiment un copier/coller mais on peut s'en servir comme… Ces commandes copient ou coupent une zone rectangulaire.

- CopyArea()
- ClearRect()

Texte et Polices

Pour écrire du texte, on utilise la commande drawString(texte, x, y).
On peut mettre une police en état normal, italique, ou gras :

- Font.PLAIN ; //état normal
- Font.BOLD ; // état gras
- Font.ITALIC ; // état italique

Un petit exemple (encore pas tout à fait fonctionnel) :

- Font f = new Font (" TimesRoman ", Font.BOLD + Font.ITALIC, 24) ; // 24 est la taille
- Screen.setFont(f) ;
- Screen.drawString("Bonjour !", 30, 30);

Pour obtenir des informations sur du texte:

- stringWidth(String) // largeur totale de la chaîne
- charWidth(char) // largeur du caractère
- getHeight // hauteur de la police

Applet : *Floride*

```
import java.awt.Graphics;
import java.awt.Polygon;

public class Floride extends java.applet.Applet {
    public void paint(Graphics screen) {
        screen.drawString("Florida", 185, 75);
        screen.drawLine(185,80,222,80);
        screen.drawRect(2, 2, 345, 345);
        screen.drawRoundRect(182,61,43,24,10,8);
        int x[] = { 10, 234, 253, 261, 344, 336, 295, 259, 205, 211,
            195, 191, 120, 94, 81, 12, 10 };
        int y[] = { 12, 15, 25, 71, 209, 278, 310, 274, 188, 171, 174,
            118, 56, 68, 49, 37, 12 };
        int pts = x.length;
        Polygon poly = new Polygon(x, y, pts);
        screen.drawPolygon(poly);
        screen.fillOval(235,140,15,15);
        screen.fillOval(225,130,15,15);
        screen.fillOval(245,130,15,15);
        for (int ax = 50; ax < 150; ax += 10)
            for (int ay = 120; ay < 320 ; ay += 10)
                screen.drawArc(ax, ay, 10, 10, 0, -180);
    }
}
```

Une petite explication : les codes de ce chapitre n'étaient pas fonctionnels car ils n'étaient pas dans la méthode paint() de l'applet. Ici la méthode paint est déclarée avec l'objet Graphics nommé *screen*, on aurait pu l'appeler *g*. *screen* doit être mis avant chaque instruction qui trace une ligne ou chaîne.

Le fichier class généré par la compilation doit être lancé par l'intermédiaire d'un navigateur compatible Java 2 (disons que l'appletviewer et toujours la meilleure alternative), avec le code HTML suivant :

```
<body bgcolor="#c4c4c4">
<div align="center">
<applet code="Floride.class" height=350 width=350>
</applet>
</div>
</body>
```

Cet exemple est tiré de l'excellent livre JAVA 2 de Roger Cadenhead & Laura Lemay aux éditions Campus Press (titre original : Teach yourself Java in 21 days).