

Interface graphique

PO2 (2014-2015)

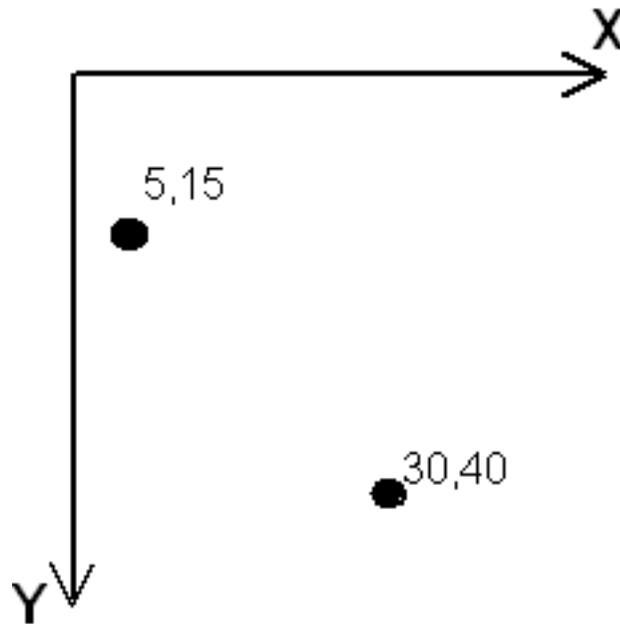
Chapitre 5 Dessins

Les principes de base du dessin

- Système de coordonnées
- Graphics
- Color

Système de coordonnées

- Le système de coordonnées de Java utilise le pixel comme unité de mesure. Toutes les valeurs de pixels sont des entiers.



Systeme de coordonnees

- Le point d'origine (0,0) correspond à l'angle supérieur gauche du panneau. La valeur de la coordonnée x croît à mesure que l'on s'éloigne vers la droite du point (0,0) et la coordonnée y croît à mesure que l'on s'éloigne vers le bas par rapport au point (0,0).

Graphics

- Pour dessiner, nous utilisons la classe `java.awt.Graphics`, qui contient des méthodes permettant de dessiner du texte ou des formes sur un composant `JPanel`

Graphics

- Segments
drawLine(int, int, int, int)
- Rectangles
drawRect(int, int, int, int) et fillRect(int, int, int, int)
- Ellipses et cercles
drawOval(int, int, int, int) et fillOval(int, int, int, int)
- Polygones
drawPolygon(int[], int[], int) et fillPolygon(int[], int[], int)
- Arcs
drawArc(int, int, int, int, int, int) et fillArc(int, int, int, int, int, int)

Graphics

- Deux façons de dessiner une figure:
 - Récupérer l'objet Graphics avec le méthode `getGraphics()` de l'objet JPanel
 - Redéfinir la méthode `paintComponent(Graphics g)` de JPanel, et utiliser `repaint()` pour obliger le composant JPanel à se redessiner

Color

- La classe Color encapsule des méthodes et des constantes pour gérer les couleurs
- Constructeurs :
 - Color(int r, int g, int b)
 - Crée une couleur avec les proportions de rouge (r), de vert (g) et de bleu (b) spécifiées. Chacun de ces trois nombres doit être compris entre 0 et 255
 - Color.WHITE, Color.BLACK, etc

Color

- Lorsqu'on dessine sur un composant, le composant lui attribue une couleur, et cette couleur peut être modifiée par la méthode :
 - `setColor(Color)` : fixer une nouvelle couleur de dessin

Exemple : DePanel.java

```
public void paintComponent(Graphics g) {
    super.paintComponent(g);
    int w = getWidth();
    int h = getHeight();
    g.setColor(Color.red);
    g.drawRect(0, 0, w-1, h-1);
    switch (val) {
        case 1: drawSpot(g, w/2, h/2);
                break;
        case 3: drawSpot(g, w/2, h/2);
                // go to next case
        case 2: drawSpot(g, w/4, h/4);
                drawSpot(g, 3*w/4, 3*h/4);
                break;
        case 5: drawSpot(g, w/2, h/2);
                // go to next case
        case 4: drawSpot(g, w/4, h/4);
                drawSpot(g, 3*w/4, 3*h/4);
                drawSpot(g, 3*w/4, h/4);
                drawSpot(g, w/4, 3*h/4);
                break;
        case 6: drawSpot(g, w/4, h/4);
                drawSpot(g, 3*w/4, 3*h/4);
                drawSpot(g, 3*w/4, h/4);
                drawSpot(g, w/4, 3*h/4);
                drawSpot(g, w/4, h/2);
                drawSpot(g, 3*w/4, h/2);
                break;
    }
}
```

```
private void drawSpot(Graphics g, int x, int y) {
    g.fillOval(x-SPOT_DIAMETER/2, y-SPOT_DIAMETER/2,
    SPOT_DIAMETER, SPOT_DIAMETER);
}
}
```