# Partie I : Programmation graphique en Swing

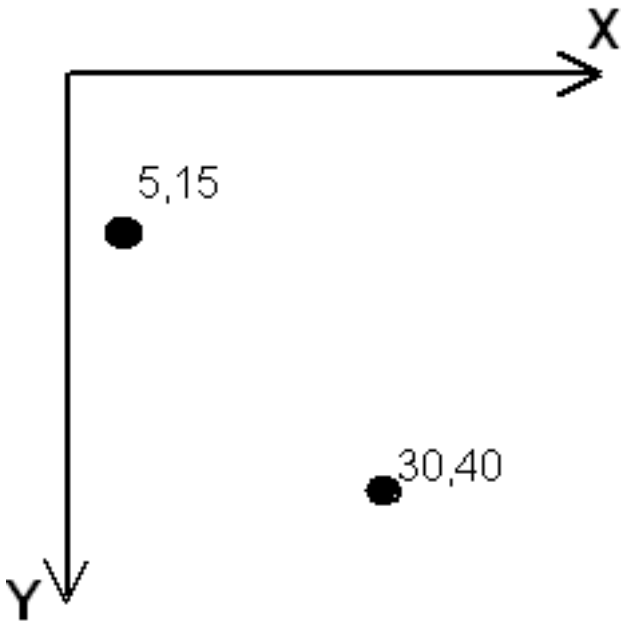
Programmation objet 2 (2012-2013)

Chapitre 6 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.



# Système de coordonnées

- 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.
- Pour dessiner sans effacement, il faut redéfinir la méthode paintComponent() du composant.
- Pour obliger un composant à se redessiner, utiliser repaint().

# Graphics

- La classe Graphics permet de dessiner et de remplir des figures ayant des formes élémentaires
  - 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

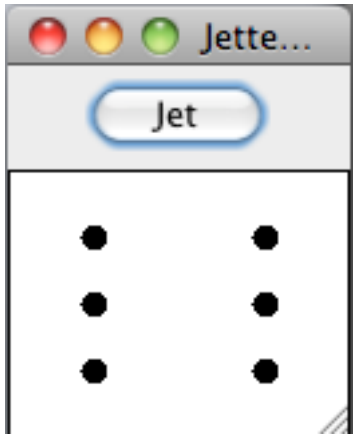
- Lorsqu'on dessine sur un composant, le composant lui attribue une couleur, et cette couleur peut être modifiée par la méthode :
  - setColor(Color) : fixer une nouvelle couleur de dessin

# 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
  - Etc.

# Exemple
## MVC1/VueGraphique.java

```
public class VueGraphique extends JPanel implements
Observer {
   private static final int SPOT_DIAMETER = 10;
   private int val;
   public VueGraphique() {
        setBackground(Color.white);
        setPreferredSize(new Dimension(100,100));
   }
   public void update(Observable o, Object arg) {
                Integer valObj = (Integer) arg;
                val = valObj.intValue();
                repaint();
   }
```



```
public void paintComponent(Graphics g) {
      super.paintComponent(g);
      int w = getWidth();
      int h = getHeight();
      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);
   }
}
```