

Introduction à Java2D

Introduction Java 2D

- Il est utile de dessiner son propre composant ou de créer un panel graphique afin d'afficher des figures géométriques, des graphes, créer un jeu vidéo,...
- Java 2D: Ensemble de classes destinées à la gestion de l'affichage 2D sous Java 2 et la manipulation de:
 - graphiques vectoriels,
 - images bitmap,
 - texte et polices de caractères,
 - fonctions d'impression,

L'affichage en Java

- Les classes développées et qui héritent de JComponent seront obtenues par dérivation et surcharge de la méthode ***void paint(Graphics g)*** pour la gestion de leurs affichages.
- Méthode appelée automatiquement en cas de détection d'un événement entraînant un affichage d'un objet.
- ***Affichage en java 2D***: utilisation de la classe **Graphics2D** dérivant de Graphics et apportant des fonctionnalités supplémentaires tout en assurant une compatibilité ascendante grâce à la dérivation.

Fonctionnalités supplémentaires

- **Système de coordonnées:** l'objet **Graphics2D** contient un objet qui définit *AffineTransform* comment convertir des coordonnées UserSpace vers leur équivalent dans le repère d'affichage.
- **Affichage plus évolué:** Java2D apporte des améliorations pour l'affichage graphique par rapport aux versions précédentes :
 - Formes évoluées: Shape
 - Contrôle de l'outil de dessin : Anti-crénelage (anti-aliasing)
 - Pinceau pour le bord et l'intérieur des surfaces
 - Clipping
 - Composition de pixels, d'images
 - Optimisation de la vitesse ou de la qualité d'affichage

Fonctions de Graphics2D

Les graphiques vectoriels

- **Objets vectoriels**
 - Segments de droites:
 - **void drawLine(int x1, int y1,int x2,int y2);**
 - Rectangles remplis ou non
 - **void drawRect(int x,int y,int width,int height);**
 - **void fillRect(int x,int y,int width,int height);**
 - Cercles et ellipses remplis ou non:
 - **void drawOval (int x,int y,int width,int height);**
 - **void fillOval (int x,int y,int width, int height);**

Fonctions de Graphics2D

Les graphiques vectoriels

```
public void paintLine(Graphics g,int x1,int y1,int x2,int y2){  
    Graphics2D g2 =(Graphics2D) g;  
    g2.drawLine(x1,y1,x2,y2) ;  
}  
public void paintRect(Graphics g,int x,int y){  
    Graphics2D g2 =(Graphics2D) g;  
    g2.fillRect(x,y,120,30) ;  
}
```

Fonctions de Graphics2D

Les graphiques vectoriels (2)

- **Objets vectoriels**
 - Polygones remplis ou non:
 - **void drawPolygon**(int[] xPoints,int[] yPoints,int nPts);
 - **void fillPolygon**(int[] xPoints,int[] yPoints, int nPts);
 - Formes évoluées remplis ou non:
 - **void draw(Shape s);**
 - **void fill(Shape s);**

Fonctions de Graphics2D

Les graphiques vectoriels (2)

- ***Les images bitmap et la gestion des pixels***
 - Fonctions d'affichage des images:
 - **boolean drawImage**(Image img,int x,int y,Color bgcolor, ImageObserver observer);
 - **boolean drawImage**(Image img,int x,int y, ImageObserver observer);

Quelques Fonctions de Graphics2D: Texte et Polices

- **Affichage de texte**
 - ***Fonctions de la classe Graphics:***
 - void drawString(AttributedCharacterIterator ite, int x,int y);
 - void drawString(String str,int x,int y);
 - ***Fonctions de la classe Graphics2D:***
 - void drawString(AttributedCharacterIterator ite,int x,int y);
 - void drawString(String s,int x,int y);

Quelques Fonctions de Graphics2D: Texte et Polices

- **Fonction de la classe TextLayout:**
 - void draw(Graphics2D g2, float x, float y);TextLayout permet une large part des manipulations sur les polices de caractères.
- **Choix de la police de caractères**
 - Fonctions de la classe Graphics:
 - void setFont(Font font);
- **Liste des polices de caractères du système**
 - Fonctions de la classe GraphicsEnvironment:
 - String[] getAvailableFontFamilyNames();
 - Font[] getAllFonts();
 - GraphicsEnvironment
getLocalGraphicsEnvironment();

Exemple de manipulation de textes

```
public void paintText(Graphics g,int x,int y) {  
    Graphics2D g2 =(Graphics2D) g;  
    g2.setColor(Color.red);  
    g2.drawString("Caractères simple colorée en rouge",x+50,y+25);  
    Font f1 = new Font("Comic Sans MS",Font.BOLD,22);  
    g2.setFont(f1);  
    g2.setColor(Color.magenta);  
    g2.drawString("Fonte : 22 Comic Sans MS Bold",30+x,55+y);  
    AffineTransform at = new AffineTransform();  
    at.setToRotation(0.05);at.setToTranslation(10,0);  
    g2.setTransform(at);  
    g2.setColor(Color.blue);  
    Font f2 = f1.deriveFont(Font.ITALIC|Font.PLAIN);  
    g2.setFont(f2);  
    g2.drawString("Fonte avec une légère rotation",30+x,85+y);  
}
```