

# Bibliographie technique

*Java 2D pour la génération d'images au format PNG*

*Documentation officielle sur Java 2D*

<http://docs.oracle.com/javase/tutorial/2d/index.html>

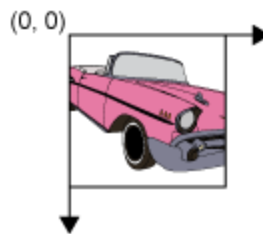
**Cette bibliographie présente les classes de java2D utiles à la réalisation du projet**

## **Présentation de l'API Java 2D**

Java 2D est une bibliothèque graphique java qui vient en extension de l'AWT qui présentait des manques et des déficiences. Les classes de Java 2D ne sont pas regroupées en un package unique mais ajoutées aux classes préexistantes de l'AWT (Abstract Windows Toolkit).

## **Système de coordonnées**

L'origine du repère est en haut à gauche de la fenêtre. Les coordonnées sont des nombres entiers qui correspondent à un nombre de pixels.



## **La classe Font**

Cette classe contient toutes les informations nécessaires pour appliquer une police de caractères à une chaîne.

Un constructeur de la classe Font est : *Font(String name, int style, int size)*

## **BufferedImage**

Dans l'API Java <sup>™</sup> une image 2D est une matrice bidimensionnelle rectangulaire de pixels, où chaque pixel représente la couleur à cette position de l'image et où les dimensions représentent la largeur et la hauteur de l' image telle qu'elle est affichée.

La classe la plus importante pour représenter ces images est `java.awt.image.BufferedImage`. L'API Java 2D stocke le contenu de ces images en mémoire afin qu'elles puissent être directement accessibles. La classe `BufferedImage` fournit des méthodes pour lire et écrire des images dans la mémoire tampon.

// Créer une image en mémoire tampon à partir des images types prédéfinis :  
*`new BufferedImage(width, height, type, colorModel)`*

## **Graphics 2D**

La classe `BufferedImage` ne fournit pas directement des méthodes de dessin. Il faut créer un objet `Graphics` avec la méthode `BufferedImage.createGraphics()`

//Créer un objet `Graphics2D` qui peut être utilisé pour écrire dans l'objet `BufferedImage` :  
*`createGraphics()`*

Plusieurs méthodes de la classe `graphics2D` sont intéressantes :

Appliquer la police de caractères au composant :  
`setFont(Font font)`

Positionner la chaîne de caractères passée en paramètre sur le composant `graphics2D` :  
`drawString(String str, float x, float y)`

## **Point 2D**

La classe `Point` crée un point représentant un emplacement dans l'espace de coordonnées (x,y). Les sous-classes `Point2D.Float` et `Point2D.Double` permettent d'utiliser des coordonnées respectivement de type `float` et `double`.

// Création `Point2D.Double`  
`Point2D.Double points = new Point2D.Double (x, y);`

En outre la classe Point2D a des méthodes de calcul de distances.

## Ellipse 2D

Créer une ellipse définie par un rectangle.

```
new Ellipse2D.Double(double x, double y, double w, double h)
```

## Premiers tests

Grâce à cette librairie, nous avons pu produire l'image présentée ci-dessous. Elle représente les films sur une carte à partir des coordonnées fournies en entrée.

