

# TP4 Programmation Java+Graphics

Dans ce TP, nous allons utiliser le mode graphique le plus simple de Java : celui de AWT. Attention, il n'a pas conseillé de l'utiliser autrement que pour ce TP. De bien meilleurs modes existent (certains de base dans Java comme Swing) ou en bibliothèques extérieures (JavaFX, voir plus loin). Il existe aussi des bibliothèques (API) dédiés (souvent payantes) pour les jeux *etc.*

## Exemple :

```
import java.awt.*;
import java.awt.event.*;

public class dessin1 extends Canvas {
    public void update(Graphics g) {paint(g);}

    // ICI LE CODE A MODIFIER POUR LES EXERCICE
    //*****
    public void paint(Graphics g) {
        g.setColor(Color.BLUE);
        g.fillRect(0,0, 100, 100);
        g.setColor(Color.GREEN);
        g.drawString("Coucou (vert+rouge=jaune)", 10, 60);
        System.out.println("coucou");
    }
    //*****

    public static void main(String[] arg) {
        Frame f = new Frame(); // Créer une fenêtre
        f.setBounds(0, 0, 500, 500); // Lui donner une dimension
        f.add(new dessin1()); //Mettre le programme de dessin
        //Faire fonctionner la « croix » pour quitter l'application
        f.addWindowListener( new WindowAdapter() {
            public void windowClosing(WindowEvent ev) {
                System.exit(0);
            }
        });
        // Montrer le tout
        f.show();
    }
}
```

Vous trouverez pleins d'autres fonctionnalités dans :

<https://docs.oracle.com/javase/10/docs/api/java/awt/Graphics.html>

Exemples :

```
g.drawOval(100,100, 100,100);
g.drawLine(100, 100, 200, 200);
getSize().width => donne la largeur de la fenêtre
getSize().height => donne la la hauteur
```

Exemple a tester :

```
int t=140;
g.setColor(Color.RED);
for (int i=0; i<=t; i+=5) g.drawLine(100, 100+i, 100+i, 100+t);
```

**Exercice 1.1** Testez le code ; Bougez la fenêtre puis Redimensionnez là et en déduire le comportement de `update`.

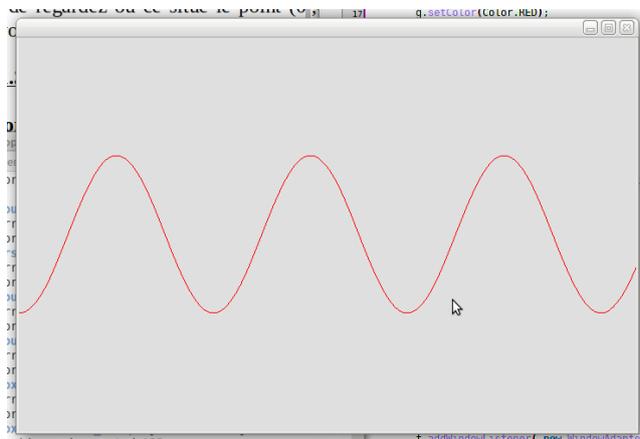
**Exercice 1.2** Remplacez le code de `paint()` par le petit exemple ci-dessus. Testez.

Je conseil de regardez ou ce situe le point (0 ,0) ainsi que le point (100,100)...Utilisez une ligne pour le savoir.

**Exercice 1.3** Dessinez une maison (un carré avec un triangle au dessus)

```
n = (int)(Math.random() * 5); //génère un nombre aléatoire compris entre 0 et 5.  
g.setColor(new Color(1,2,3)) ; //change la couleur du stylo avec 1 de rouge, 2 de vert et  
3 de bleu (RGB)
```

**Exercice 1.4** Tracez aléatoirement 100 rectangles de tailles aléatoires de couleurs aléatoires. Faites un mélange rectangles/cercles. Amusez vous un peu !



**Exercice 1.6** Tracez la fonction cosinus un peu comme dans l'exemple ci-dessous. Testez d'autres fonction (sinus, des fonctions de votre imagination ou déjà vu en cours).

**Information :** `Math.round(5.78)` donne un int à partir d'un double en prenant la valeur la plus proche. On peut aussi écrire `(int)5.78`. `Math.cos(5.67)` donne le cosinus. Et on peut aussi écrire `(double)5` pour convertir un entier en double.

**Exercice 1.7**

En rajoutant `import java.util.Arrays;` comme on rajoute pour le Scanner, on peut transformer un tableau en un flot (stream) et vice-versa. Exemple :

```
int[] tabX = {1,2,3,4,5,6,7,8};  
int[] tabY = Arrays.stream(tabX).map(x -> x*x).toArray();  
Arrays.stream(tabY).forEach(x -> {System.out.println(x); });
```

Ce code transforme le tableau X en flot puis applique la fonction carré puis retransforme en tableau. Puis on affiche. Écrivez une méthode qui prend en paramètre un tableau de floats et affiche en mode graphique le sinus des valeurs du tableau comme pour l'exercice 6.

# JAVAFX (pour Java >11)

Pour ceux qui ont réussi à installer Java11, je présente ici très brièvement une API moderne, efficace et pratique pour faire des applications Java IHM/graphiques (IHM=Interface Homme Machine = les widgets = boutons, fenêtres, menus *etc.*)

## Installation :

Téléchargez la version 11.0.2 SDK sur <https://gluonhq.com/products/javafx/>  
Décompressez et mettez dans le dossier de votre choix.

## En ligne de commande :

LINUX/Apple export PATH\_TO\_FX=path/to/javafx-sdk-11.0.2/lib  
Windaube set PATH\_TO\_FX="path\to\javafx-sdk-11.0.2\lib"

## Pour compiler :

LINUX/Apple javac --module-path \$PATH\_TO\_FX --add-modules javafx.controls prog.java  
Windaube javac --module-path %PATH\_TO\_FX% --add-modules javafx.controls prog.java

## Pour exécuter :

LINUX/Apple java --module-path \$PATH\_TO\_FX --add-modules javafx.controls prog  
Windaube java --module-path %PATH\_TO\_FX% --add-modules javafx.controls prog

Installation pour ECLIPSE : <https://openjfx.io/openjfx-docs/#install-javafx>

Documentation: <https://openjfx.io/javadoc/11/>

## Exemple :

```
import javafx.application.Application;
import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.scene.Group;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.paint.Color;
import javafx.stage.Stage;
import javafx.scene.control.Label;
import javafx.scene.layout.StackPane;
import javafx.scene.layout.FlowPane;
import javafx.scene.shape.Circle;

public class fx extends Application {

    public static void main(String[] args) {
        launch();
    }

    // Ici on mettra « tout » l'interface graphique
    public void start(Stage primS) {
        String javaVersion = System.getProperty("java.version");
        String javafxVersion = System.getProperty("javafx.version");
```

```

    primS.setTitle("Mon premier programme Java "+javaVersion+" avec JavaFX
"+javafxVersion+" !");
    //Label l = new Label("Hello world!");
    // Rajouter un placement automatique des widgets ; Meme en cas de redimension
    FlowPane SP = new FlowPane();
    Scene scene = new Scene(SP, 430, 100);
    // Ou un placement manuel complet...
    //Group root = new Group();
    //Scene scene = new Scene(root, 300, 250, Color.LIGHTGREEN);
    // Rajouter un bouton
    Button btn = new Button("Cliquez Ici");
    // Une fonctionnelle pour dire l'action du bouton (voir cours Ing1)
    btn.setOnAction(e -> {System.out.println("Hello World");});
    // On rajoute le bouton à la scene (fenetre en gros)
    SP.getChildren().add(btn);

    // On dessine maintenant
    Circle cercle = new Circle();
    cercle.setCenterX(400);//réglage de la position, de la taille et de la couleur du cercle
    cercle.setCenterY(200);
    cercle.setRadius(100);
    cercle.setFill(Color.BLUE);
    cercle.setStroke(Color.ORANGE);//réglage de la couleur de la bordure et de son épaisseur
    cercle.setStrokeWidth(5);

    Circle cercle2 = new Circle();
    cercle2.setCenterX(600);//réglage de la position, de la taille et de la couleur du cercle
    cercle2.setCenterY(400);
    cercle2.setRadius(100);
    cercle2.setFill(Color.YELLOW);
    cercle2.setStroke(Color.ORANGE);//réglage de la couleur de la bordure et de son épaisseur
    cercle2.setStrokeWidth(5);

    SP.getChildren().add(cercle);//on ajoute le cercle au dessin
    SP.getChildren().add(cercle2);//on ajoute le cercle au dessin

    // Affichage !
    primS.setScene(scene);
    primS.show();
}
}

```

**Exercice 1.7** Refaire les précédents exercices avec JavaFX.