Correction TP3

Le Texte

```
package com.mco;

import javafx.application.Application;
import javafx.scene.Group;
import javafx.scene.Scene;
import javafx.scene.paint.Color;
import javafx.scene.text.*;
import javafx.stage.Stage;

public class Main extends Application {
    public static void main(String[] args) {
        System.out.println("Polices disponibles :");
        for (String fontName: Font.getFamilies()) {
            System.out.println(fontName);
        }

        launch(args);
    }


    public void start(Stage theStage) {
        theStage.setTitle("Hello, World!");

        //mise en place du Stage : création du scene graph et du noeud
racine.
        Group root = new Group(); //ce ne sera pas une feuille, donc de
classe Group.
        Scene theScene = new Scene(root); //création du scene graph
        theStage.setScene(theScene); //on le place dans le stage

        //Création d'un text
        Text text = new Text();

        Font comic = new Font("Comic sans MS", 50); //une nouvelle police de
taille 50
        text.setFont(comic);
        text.setFill(Color.color(.5,0,1.0,1.0));

        text.setText("Hello World aussi !");
        text.setX(0);
        text.setY(200);

        //on l'ajoute à notre scene graph via son père, ici le noeud root
        root.getChildren().add(text);
```

```
        theStage.show();

    }
}
```

Des ronds

```
gc.setFill(Color.rgb(200, 0, 0, .5));
gc.fillOval(100, 200, 150, 150);

gc.setFill(Color.rgb(0, 200, 0, .5));
gc.fillOval(100, 100, 150, 150);

gc.setFill(Color.rgb(0, 0, 200, .5));
gc.fillOval(200, 100, 150, 150);

gc.setFill(Color.rgb(200, 0, 200, .5));
gc.fillOval(200, 200, 150, 150);
```

Afficher le Delta

Main.java

```
package com.mco;

import javafx.application.Application;
import javafx.scene.Group;
import javafx.scene.Scene;
import javafx.scene.text.Text;
import javafx.stage.Stage;

public class Main extends Application {

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

    @Override
    public void start(Stage primaryStage) {
        primaryStage.setTitle("Le temps");

        Group root = new Group();
        Scene theScene = new Scene(root, 100, 200); //taille initiale.
        primaryStage.setScene(theScene);
```

```
        Text text = new Text();
        root.getChildren().add(text);
        text.setX(0);
        text.setY(100);

        Loop loop = new Loop(text);

        loop.start();
        primaryStage.show();
    }
}
```

Loop.java

```
package com.mco;

import javafx.animation.AnimationTimer;
import javafx.scene.text.Text;

public class Loop extends AnimationTimer {

    private Text text;

    private long previousTime;

    public Loop(Text text) {
        this.text = text;
        this.previousTime = System.nanoTime(); //évite les effet de bord à
l'initialisation de l'application
    }

    @Override
    public void handle(long now) {
        long delta = now - previousTime;
        previousTime = now;

        text.setText(String.valueOf(delta));
    }

}
```

La petite aiguille

```
package com.mco;

import javafx.animation.AnimationTimer;
import javafx.scene.canvas.Canvas;
import javafx.scene.canvas.GraphicsContext;
```

```
import javafx.scene.paint.Color;


public class Loop extends AnimationTimer {
    private Canvas canvas;

    final double ROTATION  = (2 * Math.PI) * 1E-9; // une rotation par
seconde
    final Color BACKGROUND_COLOR = Color.ALICEBLUE;

    final Color WATCH_HAND_COLOR  = Color.BISQUE;
    final int WATCH_HAND_LINE_WIDTH = 10;
    final double WATCH_HAND_LENGTH = 200;

    private double angleBig;
    private double angleSmall;
    private long currentTime;

    public Loop(Canvas canvas) {
        currentTime = System.nanoTime();
        this.canvas = canvas;
    }

    @Override
    public void handle(long now) {
        handleTime(now);
        drawStuff();
    }

    private void handleTime(long now) {
        changeAngle(now - currentTime);
        currentTime = now;
    }

    private void changeAngle(long delta) {
        angleBig += ROTATION * delta;
        angleBig %= 2 * Math.PI;

        angleSmall += (ROTATION / 12) * delta;
        angleSmall %= 2 * Math.PI;
    }

    private void drawStuff() {
        clearCanvas();
        drawWatchHand();
    }

    private void clearCanvas() {
        GraphicsContext gc = canvas.getGraphicsContext2D();

        gc.setFill(BACKGROUND_COLOR);
```

```
        gc.fillRect(0, 0, canvas.getWidth(), canvas.getHeight());
    }

    private void drawWatchHand() {
        GraphicsContext gc = canvas.getGraphicsContext2D();

        double middleX = canvas.getWidth() / 2;
        double middleY = canvas.getHeight() / 2;

        gc.beginPath();
        gc.setLineWidth(WATCH_HAND_LINE_WIDTH);
        gc.setStroke(WATCH_HAND_COLOR);

        gc.moveTo(middleX, middleY);
        gc.lineTo(middleX + WATCH_HAND_LENGTH * Math.cos(angleBig),
                  middleY + WATCH_HAND_LENGTH * Math.sin(angleBig));

        gc.moveTo(middleX, middleY);
        gc.lineTo(middleX + (WATCH_HAND_LENGTH / 4) * Math.cos(angleSmall),
                middleY + (WATCH_HAND_LENGTH / 4) * Math.sin(angleSmall));

        gc.stroke();
    }
}
```