

# NFA032: projet

Année 2018/2019, semestre 2

## 1 Objectif du projet

Le projet consiste à réaliser un logiciel qui lit puis transforme des données essentiellement numériques pouvant correspondre à des mesures, sondages, etc., en diagrammes ou graphiques les rendant plus faciles à visualiser. Plusieurs types de diagrammes (au moins deux) seront proposés par votre logiciel et divers paramètres pourront être spécifiés par l'utilisateur : les couleurs, la taille du graphique, etc.

Les données seront lues à partir d'un fichier CSV qui contiendra une colonne d'entête et une ou plusieurs colonnes de données (nombres à virgule). La première ligne sera une ligne d'entête. Ces fichiers CSV sont faciles à créer avec un tableur (Excel, LibreOffice, Google Sheets, etc) ou directement avec un éditeur.

Les types de graphiques seront certains de ceux proposés par les tableurs : graphiques à colonnes, graphiques en courbes, graphiques à secteurs, graphiques à barres, etc.

Le graphique réalisé aura la forme d'un fichier SVG (Scalable Vector Graphics), que nous décrivons dans la partie 3.

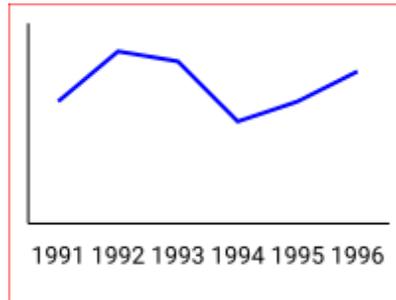
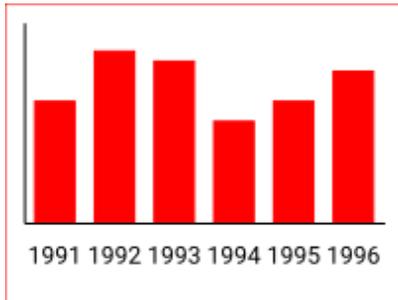
## 2 Version de base

Dans la version de base, on ne considérera que les deux premières colonnes des données : la première colonne comprendra des labels et la seconde des données numériques.

Prenons un exemple de fichier CSV à deux colonnes :

```
année, pommes
1991, 12
1992, 17
1993, 16
1994, 10
1995, 12
1996, 15
```

Et voici deux exemples de graphiques (graphique à colonne et graphique en courbe) traduisant ces données :



Ces deux types de graphiques montrent des exemples de ce que votre programme peut réaliser comme transformation de données. Il sont plus faciles à réaliser que, par exemple, les graphiques à secteurs (*camemberts*).

Le programme à réaliser devra commencer par lire des données à partir d'un fichier CSV, proposer les différents types de graphique pour la transformation et un fois un type choisi, il donnera la possibilité de modifier différents paramètres avant de créer (et enregistrer) un fichier contenant l'image au format SVG. Votre programme ne prendra pas en charge la visualisation des images SVG. Elle pourra être réalisé en externe par des outils ou navigateurs trouvables sur internet, par exemple, <https://framavectoriel.org> qui est un outil libre pour la création et visualisation du format SVG.

### 3 Le format SVG

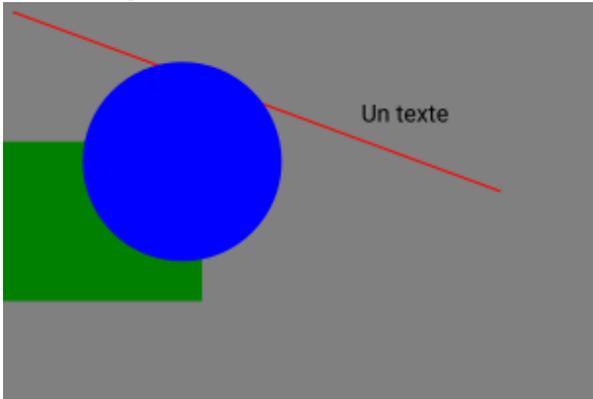
Le format SVG est un format de fichier standard utilisé notamment dans des pages web. C'est un format textuel assez facile à lire et à écrire. Il est basé sur XML : c'est un langage à balises avec des enregistrements et des attributs.

Voyons un petit exemple de fichier svg :

```
<?xml version="1.0" encoding="utf-8"?>
<svg xmlns="http://www.w3.org/2000/svg" version="1.1"
      width="300" height="200">
  <title>Exemple simple de figure SVG</title>
  <desc>
    Cette figure est constituée d'un rectangle,
    d'un segment de droite et d'un cercle.
  </desc>

  <rect width="300" height="200" x="0" y="0" fill="grey" />
  <rect width="100" height="80" x="0" y="70" fill="green" />
  <line x1="5" y1="5" x2="250" y2="95" stroke="red" />
  <circle cx="90" cy="80" r="50" fill="blue" />
  <text x="180" y="60">Un texte</text>
</svg>
```

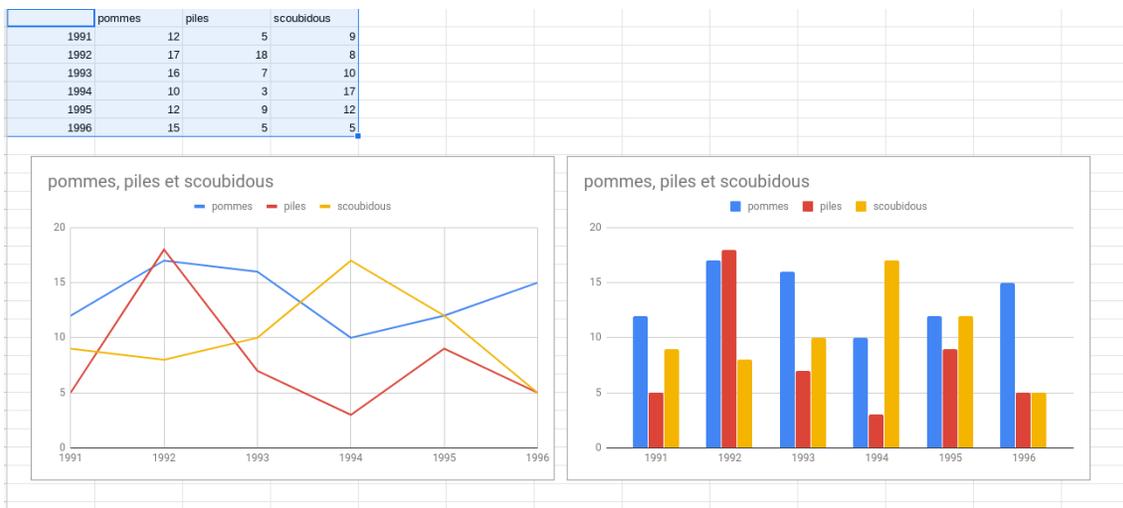
L'image décrite est la suivante :



Sur la troisième ligne, `width="300"` (largeur) et `height="200"` (hauteur) définissent la taille de l'image. Ensuite, le titre et la description sont des commentaires qui ne changent rien à ce qu'on voit. Après la ligne blanche, on voit un premier rectangle (mot-clé `rect`) qui définit la couleur du fond de l'image. Ce rectangle commence au point  $x=0$ ,  $y=0$ , qui est le coin en haut à gauche de l'image et il a la même taille que l'image. L'attribut `fill` définit la couleur du rectangle (`fill="grey"`). La ligne suivante comporte un rectangle plus petit et de couleur verte (`fill="green"`). Son coin haut gauche est le point  $x=0$ ,  $y=70$ . La ligne suivante définit un segment de droite (`line`) avec les deux points à ses extrémités. L'attribut `stroke` définit la couleur du segment. La ligne suivante est un cercle bleu avec son centre et son rayon. Ensuite vient du texte. On voit que les éléments sont affichés dans l'ordre et c'est ainsi que le cercle se superpose par-dessus la ligne et le rectangle définit avant lui.

## 4 Version avancée

Dans une version avancée du projet (en option, pas obligatoire), le graphique représentera plusieurs colonnes de données. On aura alors des graphiques analogues à ceux qui suivent.



## 5 Qualités attendues pour votre programme

Il doit être écrit dans le style orienté objet, avec un découpage en classes rationnel. Les méthodes doivent être incluses dans les bonnes classes. Les méthodes ne doivent pas être trop longues. Les erreurs doivent être gérées avec des exceptions. Les données doivent être bien protégées et les objets garder leur cohérence.

Vous représenterez les différentes données au moyen de classes différentes en utilisant l'héritage ou des interfaces (éventuellement les deux).

Le programme doit être bien présenté, avec une indentation correcte.

Le programme doit montrer vos capacités à programmer. Pour cela, il faut privilégier les classes et méthodes que vous écrivez vous-même par rapport aux classes et méthodes de la librairie. Vous pouvez utiliser les classes de `java.util` (par exemple `ArrayList`) quand même.

Le programme devra être divisé en méthodes de taille raisonnable (maximum une page d'écran). Cela vaut aussi pour la méthode `main` qui ne doit pas être trop longue.

## 6 Consignes

Le projet est individuel : chaque élève doit le réaliser et doit en écrire seul chacune des lignes de code. Le projet donne lieu à une soutenance au cours de laquelle l'élève doit justifier les choix qu'il a fait.

Pendant la période de préparation du projet, vous pouvez demander l'aide de votre tuteur. Vous pouvez lui poser toutes les questions et lui soumettre tous vos problèmes.

## 7 Remise et soutenance du projet

Il y aura trois étapes dans l'évaluation du projet. Les dates seront précisées prochainement.

Première étape : le point à mi-pacours pour voir où vous en êtes.

Deuxième étape : la remise du programme complet. Vous remettrez le code source d'un programme compilé et testé.

Troisième étape : la soutenance, en face-à-face avec un enseignant. Chaque enseignant concerné précisera les modalités pratiques (prise de rendez-vous ou file d'attente). Pour les élèves inscrits dans un centre CNAM de province, la soutenance peut se faire en visio-conférence.