

## 1 Préliminaires

Nous allons développer une application avec affichage graphique et sauvegarde des données, reprenant le travail que nous avons accompli dans le TD 8.

1. Commencer par créer les classes `Histogramme` et `Intervalles` comme vue en TD. L'histogramme comprendra 10 bins.
2. Créer une barre d'outils dans la fenetre principale de l'application avec un menu `File` avec des items `Open`, `Save`, `Clear`, `Restore`.
3. Créer une barre de d'état (status bar de type `toolStripStatus`) qui affichera respectivement "Histogram opened!", "Histogram saved!", "Histogram cleared!" et "Histogram restored!" lors de la sélection des items de menu précédents.

## 2 Chargement de l'histogramme par fichier

Ecrire la méthode pour charger l'histogramme à partir du fichier `histo_ini.dat` disponible sous Moodle. Ce fichier comprend 10 lignes, la premiere colonne est l'indice du bin et la deuxieme colonne sa valeur. L'histogramme devra s'adapter aux redimensionnements de la fenêtre et sera tracé en rouge (voir figure 1). On utilisera la classe `OpenFileDialog` pour ouvrir une boîte de dialogue permettant de sélectionner le fichier à charger, les classes `Stream` et `StreamReader` pour lire à partir d'un fichier texte, et en enfin `ReadLine()` pour lire dans le fichier ligne par ligne.

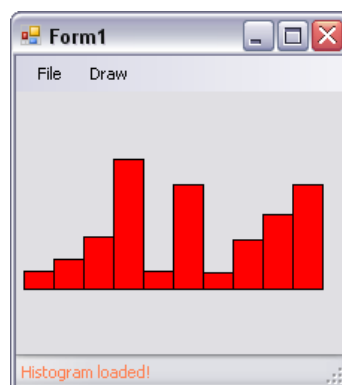


FIGURE 1 – Tracé de l'histogramme

(validation du chargement / tracé de l'histogramme et de l'implementation de la barre d'état)



## 3 Remise à zéro de l'histogramme et création par valeurs aléatoire

1. Implémenter la fonction pour que l'histogramme soit remis à zero lors de l'appui sur l'item de menu `Clear`.
2. L'histogramme pourra aussi être initialisé lors de l'appui sur la touche 'R' par tirage aléatoire d'une valeur entre 0 et 99, comme vu en TD. Implémentez cette fonctionnalité.



## 4 Serialisation de la classe Histogramme

1. En s'inspirant du TD8, créer la fonction de sérialisation permettant de sauvegarder la classe `Histogramme` dans le fichier 'histo.dat' lors du choix de l'item `Save` du menu.
2. Créer la fonction de désérialisation permettant de restaurer la classe `Histogramme` contenue dans le fichier 'histo.dat' lors du choix de l'item `Restore` du menu.



3

## 5 Tracé "Camembert" de l'histogramme

Créer un nouveau menu `Draw` comprenant les items `Bar` et `Pie`. La sélection de l'item `Bar` affichera l'histogramme comme précédemment, tandis que la sélection de l'item `Pie` l'affichera en mode "Camembert". Chaque portion du camembert sera représentative de la population d'un bin par rapport à la population totale de l'histogramme.

Pour ce faire, vous créerez :

- Une variable `total_histo` comprenant la population totale de l'histogramme.
- Un tableau d'entier `deg` contenant les angles représentant chaque bin de l'histogramme
- Un tableau de 10 couleurs, contenant les couleurs de chaque portion représentant chaque bin de l'histogramme.
- L'affichage d'une portion de camembert se fera avec la méthode `e.Graphics.FillPie(...)` ;

Vous devrez obtenir quelquechose de similaire à la figure 2 :

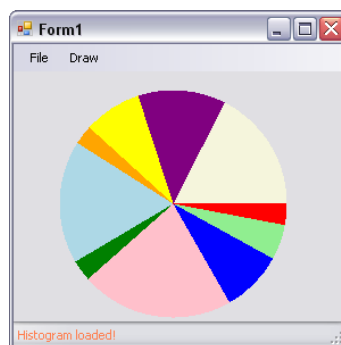


FIGURE 2 – Représentation camembert de l'histogramme



4

- Lors de l'appui sur une portion du camembert, on doublera son rayon pour mettre en valeur la portion sélectionnée.

## 6 Pour ceux qui ont fini..

- Echanger avec votre binôme voisin le fichier 'histo.dat' et essayer de restaurer la classe `Histogramme`.
- Utiliser `XmlSerializer` pour exportez vos données :

```
// Create and XmlSerializer to serialize the data to a file
XmlSerializer xs = new XmlSerializer(typeof(mon_nom_de_classe));
using (FileStream fs = new FileStream("histo.xml", FileMode.Create))
{
    xs.Serialize(fs, mon_instance_de_classe);
}
```

Idem, échanger avec votre binôme voisin le fichier 'histo.xml' et essayer de restaurer la classe `Histogramme`

### Elements de correction :

- La classe `XmlSerializer` se trouve dans le namespace `System.Xml.Serialization`.

```
System.IO.FileStream output = new System.IO.FileStream("test.dat",
    System.IO.FileMode.Create, System.IO.FileAccess.Write);
System.Xml.Serialization.XmlSerializer xs = new System.Xml.Serialization.XmlSerializer(typeof (Histogram))
```

```
xs.Serialize(output, myHisto);
```

```
output.Close();
```

- *Pour que les classes histogram et intervalle puissent être sérialisées, il faut que :*
  - *Ces classes disposent d'un constructeur par défaut (avec 0 paramètres)*
  - *Que ces classes soient publiques*