

TD d'initiation

Table of contents

1 Questions	2
2 Appendice A : comment afficher des graphes, une méthode.....	2
3 Appendice B : comment colorier des éléments.	3
4 Solution.....	3

1. Questions

1. Installez mascopt sur votre compte en suivant les instructions de la page [Getting Mascopt](#) (`../home/getting.html`) puis récupérez sur votre compte le fichier [grapheBicolore.mgl](#) (`../initiation/grapheBicolore.mgl`).
2. Écrire un programme permettant de choisir de manière graphique un fichier avec une extension `.mgl` et de stocker le graphe de ce fichier dans une variable de type `Graph`. Pour cela on utilisera la classe `GraphChooser` et la méthode `getGraphMGL` (il faut remarquer que cette méthode est utilisable ici car le fichier ne contient qu'un graphe).
3. Modifier le programme précédent de manière à construire un sous-graphe du précédent ne contenant que les sommets dont l'attribut `couleur` vaut ```rouge`". Puis un nouveau graphe contenant les sommets dont l'attribut `couleur` vaut ```vert`".
4. Modifier à nouveau le programme de façon à afficher les 2 graphes créés précédemment. Facultatif : on pourra afficher les sommets avec l'attribut `couleur` valant ```rouge`" (resp. ```vert`") en rouge (resp. vert).
5. Enfin, modifier le programme afin de sauver d'abord chaque graphe dans un fichier séparé puis tous dans un même fichier.

2. Appendice A : comment afficher des graphes, une méthode

Il existe dans `mascopt` une classe permettant l'affichage des graphes, `MascoptViewer`. Pour afficher un graphe, il faut donc créer un objet de type `MascoptViewer`. Cette objet va alors contenir une liste de vues (qui correspondent à différentes fenêtres qui seront créées) et chaque vue contient une liste de ```layers`" contenant chacun un ou plusieurs graphes. Cette organisation permet en particulier de superposer différents graphes en ajoutant leur `layers` respectives dans les vues. Il est aussi possible d'ajouter la même `layer` à différentes vues. Les vues sont implémentées par la classe `GView` et les `layers` par la classe `GLayer`.

En pratique on peut procéder de la manière suivante,

- on instancie un objet de type `MascoptViewer`

```
MascoptViewer mv = new
MascoptViewer();
```

- on instancie un objet de type `GView`

```
GView gv = mv.newViel(G.getName());
```

- on instancie un objet de type `GLayer`

```
GLayer gl = mv.newLayer(G.getName());
```

- on associe le graphe et le `layer`

```
mv.addGraphInLayer(G,gl,true);
```

- on ajoute la `layer` dans la vue

TD d'initiation

```
mv.addLayerInView(gl,gv);
```

Et tout devrait s'afficher de la manière souhaitée.

3. Appendice B : comment colorier des éléments.

Pour colorier les sommets et arêtes dans le mascotViewer, il suffit de créer un attribut ``color" dans l'élément à colorier et de lui donner comme valeur l'entier correspondant à la couleur désirée.

Ainsi, supposons que l'on souhaite voir apparaître en rouge, un façon d'aboutir à ce résultat est d'utiliser les méthodes de la classe java.awt.Color :

```
N1.setIntValue("color", (Color.red).getRGB());
```

4. Solution

Voici la solution de l'exercice, programmée par Marc Martinez: [Premier.java](#)
(../initiation/Premier.java)