



Mini projet d'Architecture Logicielle – Outils SVG

M1 Informatique

(Client : florent.nicart@univ-rouen.fr)

9 décembre 2016

1 Introduction

On souhaite disposer d'outils de manipulation de dessins au format SVG sous forme de deux programmes :

- Un visualisateur de dessin **Viewer**. Lorsqu'il est invoqué sur la ligne de commande suivi du nom d'un fichier SVG, ce dernier est affiché dans une fenêtre graphique. Si l'option `-o nomdefichier` est ajouté, l'affichage graphique est remplacé par une sortie dans un fichier au format BMP (option `-bmp`) ou bien PDF (option `-pdf`). L'afficher se comporte alors comme un convertisseur. Exemples d'utilisation :

```
> java Viewer hello.svg % Affichage  
> java Viewer hello.svg -bmp -o hello.bmp % Conversion en bmp  
> java Viewer hello.svg -o hello.pdf -pdf % Conversion en pdf
```
- Un mini éditeur **SvgEdit** qui permet d'altérer graphiquement un dessin SVG et de le sauvegarder.

2 Spécifications techniques

2.1 Le format SVG

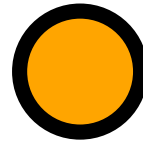
Il est bien entendu hors de question de prendre en charge l'intégralité du langage SVG. Nous considérerons donc uniquement un sous-ensemble de ce langage. Les éléments suivant seront supportés

- L'élément racine **svg** qui aura pour seul attribut reconnu la taille virtuelle du dessin (`viewBox="0 0 100 100"`).
- les éléments `<style>` qui permet d'ajouter un style de rendu nommé (format CSS). Les règles de style se limiteront à un sélecteur simple constitué du nom d'une classe (`.nom`), éventuellement le nom d'un élément (`circle`) et les attributs de rendu se limiteront à la couleur remplissage `fill`, la couleur de tracé (`stroke`) et l'épaisseur de trait (`stroke-width`). Les valeurs des couleurs se réduiront à aux constantes symboliques (*white*, *black*, *yellow*, *blue*, *red*, *orange*, ...) et l'épaisseur à la syntaxe *xpx* où *x* est un entier. Par soucis de flexibilité, lors du chargement, les éléments et les attributs non reconnus seront simplement ignorés.
- Une liste d'éléments graphiques réduites à : **line**, **circle**, **ellipse**, **rectangle** et **text**. Pour chacun de ces éléments, on ne supportera que les attributs qui lui sont spécifiques ainsi que l'attribut `class` qui sera la seule manière d'altérer les attributs de rendu (pas d'attribut `style`, pas de style en ligne).
- L'élément **g** qui permet de grouper des éléments graphiques avec pour seul attribut
`transform="translate(-26.361702,-26.787231)"`
permettant d'appliquer une translation sur le groupe.
- Voici un exemple simple :

```

1 <svg viewBox="0 0 100 100">
2   <style>
3     .rond {
4       fill : orange;
5       stroke : black;
6       stroke-width : 10px;
7     }
8   </style>
9
10  <circle cx="50" cy="50" r="40" class="rond" />
11 </svg>

```



— Voici un autre exemple avec un groupe :

```

1 <svg viewBox="0 0 100 50">
2   <style>
3     circle {
4       fill : white;
5       stroke : green;
6       stroke-width : 5px;
7     }
8   </style>
9
10  <g>
11    <circle cx="25" cy="25" r="15" />
12    <circle cx="40" cy="25" r="15" />
13    <circle cx="55" cy="25" r="15" />
14    <circle cx="70" cy="25" r="15" />
15  </g>
16 </svg>

```



Nous produirons des dessins arbitrairement complexes par combinaisons arbitraires de ces éléments.

2.2 Les formats PDF et BMP

Le format PDF est un format binaire assez proche en fonctionnalités de SVG : une page est représentée en plaçant des éléments vectoriels dans une canevas. Pour produire la sortie demandée au format PDF, on pourra s'appuyer sur une bibliothèque existante telle que *Apache PDFBox* (Voir références).

Le format BMP est un format d'image Bitmap (point à point) sans compression extrêmement simple. Il est constitué d'un en-tête court décrivant la taille de l'image et sa résolution suivi d'un tableau à deux dimensions stockant les pixels. Pour la sortie, on pourra prendre un tableau de taille arbitraire ou bien fonction de la taille de la `viewBox` et dessiner les figures points à points, puis l'enregistrer dans un fichier après l'en-tête. (Voir les références pour le détail du format)

2.3 L'éditeur

L'éditeur permet de charger un fichier SVG, de l'altérer, puis de le sauvegarder à nouveau. Les modifications peuvent être les suivantes.

- déplacer une forme sélectionnée (ou un groupe),
- supprimer une forme sélectionnée (ou un groupe),
- Ajouter une forme,
- modifier les dimensions d'une forme sélectionnée (ou d'un groupe),

L'ergonomie ne fait pas partie des aspects prioritaires du projet. Ces opérations peuvent être réalisées à l'aide de menus rudimentaires si le temps ne permet pas un interface *glisser-lâcher*.

3 Travail attendu

Le travail est à réaliser en binôme maximum et est à rendre avant le 2 janvier 2017 à 23h59. Vous fournirez une archive nommée `projet-al-nom1-nom2.zip` comportant un dossier éponyme. Celui-ci contiendra le code source de votre implémentation ainsi que votre rapport.

Il est important de noter que le rapport aura autant d'importance que l'implémentation. Celui-ci devra contenir :

- Une explication de l'architecture globale.
- Pour chaque patron mis en œuvre : la motivation du choix du patron, le diagramme UML de la partie de votre application correspondant au patron instancié (en prenant soin d'indiquer les acteurs).
- Éventuellement, une liste des patrons que vous avez hésité à utiliser et les raisons pour lesquelles vous les avez écartés.
- Le moins de fautes d'orthographe et de français possible.

Un autre point important à garder à l'esprit est que ce mini projet est un prétexte à l'élaboration d'une architecture de taille raisonnable mais respectant les bons principes vus en cours. Par conséquent :

- le logiciel n'a pas vocation à être abouti : certaines fonctionnalités énumérées peuvent ne pas être développées, en particulier lorsqu'elles sont redondantes, du moment que l'architecture est en place. On utilisera par exemple des bouchons pour les fonctions pour lesquelles le temps a manqué en l'expliquant dans le rapport.
- De même, la qualité de rendu ainsi que l'ergonomie de l'interface ne sont pas prioritaires.

Vous disposez de fichiers de test SVG dans l'archive compagne.

4 Références

- Référence en ligne du langage SVG : <https://developer.mozilla.org/fr/docs/Web/SVG>
- Apache PDFBox : <http://apache.mindstudios.com/pdfbox/2.0.3/pdfbox-2.0.3-src.zip> – voir le code `examples/src/main/java/org/apache/pdfbox/examples/pdmodel/helloworld.java`.
- Description du format BMP : https://fr.wikipedia.org/wiki/Windows_bitmap
- Le cours : [https://sympa.univ-rouen.fr/sympa/d_read/m1info/Architecture%20Logicielle/;-\)](https://sympa.univ-rouen.fr/sympa/d_read/m1info/Architecture%20Logicielle/;-))