



## PROGRAMMATION OBJET AVANCÉE ESILV S6 ~ 2015-2016

### ┌ TD Final : De csv vers svg ┐

*Ce sujet contient 6 pages. Les consignes à respecter se trouvent en page 6.*

Des dessins, utilisant des formes simples (cercle, carré, rectangle,...) sont émis par un logiciel dans le format csv (attention, le séparateur est le ;). Ce format décrit chaque forme, sa couleur, la couleur des traits, son ordre d'affichage.... Afin de les visualiser, l'objectif de ce problème est de proposer un programme permettant de convertir ces fichiers csv vers un format affichable très facilement : le svg. Un fichier d'extension .svg est un fichier texte, comme le format csv, qui contient une information structurée. Par exemple le fichier cercle.svg décrit un cercle :

---

```
<svg xmlns="http://www.w3.org/2000/svg" version="1.1">
  <circle cx="100" cy="50" r="40" style="fill:rgb(255,255,0)" />
</svg>
```

---

la première et troisième ligne sont le point d'entrée et le point de sortie du fichier svg. La seconde ligne représente un cercle, centré en (100,50) et de rayon 40. Le style permet de remplir en jaune (code rgb 255,255,0) le cercle. Vous pouvez ouvrir ce fichier avec un navigateur internet.

Un autre exemple : un cercle et un rectangle. Attention l'ordre est important : le rectangle avant le cercle ne produirait pas le même résultat. Le rectangle est de largeur 300, de hauteur 100 et est colorié en bleu (code rgb 0,0,255). Son coin, en haut à gauche, est positionné en (110,70).

---

```
<svg xmlns="http://www.w3.org/2000/svg" version="1.1">
  <circle cx="100" cy="50" r="40" style="fill:rgb(255,255,0)" />
  <rect x="110" y="70" width="300" height="100" style="fill:rgb(0,0,255)" />
</svg>
```

---

# 1- Les couleurs

Une couleur est un triplet de valeurs entières entre 0 et 255, représentant le rouge, le vert et le bleu (code RGB, cf : [http://fr.wikipedia.org/wiki/Couleurs\\_du\\_Web](http://fr.wikipedia.org/wiki/Couleurs_du_Web)).

# 2- L'ordre d'affichage

Chaque élément doit être affiché dans un ordre bien spécifique. Pour respecter cet ordre, un indice permet de repérer l'ordre d'affichage. Le fichier svg est donc construit suivant cet ordre : élément d'indice 1 en premier, d'indice 2 en second, etc...

# 3- Les éléments

Les formes représentables sont les suivantes (cf fichier ExempleTout.svg) :

- Cercle
  - Défini par un centre (cx,cy), un rayon r et une couleur R,G,B.
  - Transformations applicables : translation
  - svg : `<circle cx="..." cy="..." r="..." style="fill :rgb(...)" />`
  - csv : Cercle ;idElement ;cx ;cy ;r ;R ;G ;B ;ordre

0	1	2	3	4	5	6	7	8
Cercle	idElement	cx	cy	r	R	G	B	ordre

- Ellipse
  - Défini par un centre (cx,cy), un rayon en x rx, un rayon en y ry et une couleur R,G,B
  - Transformations applicables : translation, rotation
  - svg : `<ellipse cx="..." cy="..." rx="..." ry="..." style="fill :rgb(...)" />`
  - csv : Ellipse ;idElement ;cx ;cy ;rx ;ry ;R ;G ;B ;ordre

0	1	2	3	4	5	6	7	8	9
Ellipse	idElement	cx	cy	rx	ry	R	G	B	ordre

- Rectangle
  - Défini par le point en haut à gauche (x,y), une largeur l, une hauteur h, et une couleur R,G,B.
  - Transformations applicables : translation, rotation
  - svg : `<rect x="..." y="..." width="..." height="..." style="fill :rgb(...)" />`
  - csv : Rectangle ;idElement ;x ;y ;l ;h ;R ;G ;B ;ordre

0	1	2	3	4	5	6	7	8	9
Rectangle	idElement	x	y	largeur	hauteur	R	G	B	ordre

- Polygone Plein
  - Défini par une série de valeurs *points* et une couleur R,G,B
  - Transformations applicables : rotation
  - svg : `<polygon points="..." style="fill :rgb(...)" />`
  - csv : Polygone ;idElement ;points ;R ;G ;B ;ordre

0	1	2	3	4	5	6
Polygone	idElement	points	R	G	B	ordre

- Chemin
  - Défini par une série de valeurs *path* et une couleur R,G,B
  - Transformations applicables : rotation
  - svg : `<path d="..." style="fill :rgb(...)" />`
  - csv : Chemin ;idElement ;path ;R ;G ;B ;ordre

0	1	2	3	4	5	6
Chemin	idElement	path	R	G	B	ordre

- Texte
  - Défini par une position (x,y), une chaîne *contenu* et une couleur R,G,B
  - Transformations applicables : translation, rotation
  - svg : `<text x="..." y="..." fill="rgb(...)">...</text>`
  - csv : Texte ;idElement ;x ;y ;contenu ;R ;G ;B ;ordre

0	1	2	3	4	5	6	7	8
Texte	idElement	x	y	contenu	R	G	B	ordre

## 4- Les transformations

Deux transformations existent : la translation et la rotation. Il faut ajouter `transform` à la forme, comme illustré dans le fichier `ExempleTout.svg` et le fichier `ExempleToutSource.csv`.

- Translation

- Translate l'élément dans le plan
- Définie par deux valeurs `dx` et `dy`
- svg : Ajouter `transform="translate(dx,dy)"` à l'élément correspondant
- csv : Translation ;idElement ;dx ;dy

0	1	2	3
Translation	idElement	dx	dy

- Rotation

- Effectue une rotation centrée sur un point `(x,y)` par un angle donné
- Définie par un centre `(cx,cy)` et un angle `alpha`
- svg : Ajouter `transform="rotate(alpha x,y)"` à l'élément correspondant
- csv : Rotation ;idElement ;alpha ;x ;y

0	1	2	3	4
Rotation	idElement	angle	cx	cy

- Rotation et Translation

- Permet de coupler les deux transformations
- svg : Ajouter `transform="rotate(... .....) translate(...,...)"`
- csv : deux lignes séparées

## └ Exercice 1 ▸

Proposez une modélisation objet de l'ensemble des données. Utilisez, si nécessaire, les notions d'héritages, d'interfaces, de listes. En particulier, la classe `Dessin` (correspondant donc à un fichier) doit être présente. A quelle classe doit-on attacher la lecture csv ? l'écriture svg ? Comment assurer que les éléments soient écrits dans le bon ordre dans le fichier svg ?

## └ Exercice 2 ▸

Ecrire, sur papier, l'algorithme de lecture et de traitement d'un fichier csv.

## └ Exercice 3 ▸

Codez chacune des classes et réalisez la conversion des fichiers csv vers svg.

## └ Exercice 4 ▸

Proposez la gestion d'erreurs par exceptions.

## └ Exercice 5 ▸

Quelle(s) classe(s) est(sont) susceptible(s) d'être abstraite(s) ?

## └ Exercice 6 ▸

Documentez l'ensemble de votre code.

## Consignes

- Ce travail est à réaliser en binôme. Les binômes seront fixés au début de la séance 7.
- Ce travail est à effectuer durant les séances de TD 7/8/9. Il est à rendre au plus tard 5' après la fin de la dernière séance de TD (séance 10). Il devra être déposé sur moodle, par l'un des membres du binôme, la date dépendant de votre groupe. Tout retard entraînera des points de pénalités (-1 point par tranche de 5' de retard / au delà de 30' devoir refusé).
- Contenu de votre rendu :
  - ◇ Votre répertoire doit avoir pour nom : POA\_GRx\_NOM1\_NOM2 (avec x remplacé par votre groupe et NOM1 et NOM2 vos noms)
  - ◇ il doit contenir votre solution Visual Studio complète, nettoyée
  - ◇ il doit aussi contenir les fichier .csv à traduire (au bon endroit !)
  - ◇ **NB** il ne doit pas contenir de .svg
  - ◇ il doit aussi contenir un pdf (2-4 pages) contenant la modélisation du problème, l'algorithme demandé dans l'exercice 2, une explication rapide de votre gestion des erreurs. Si vous avez fait des choix particuliers, ou avez ajouté des éléments, vous pouvez en parler aussi.
  - ◇ vous devez ensuite compresser votre répertoire (en zip) et déposer sur moodle, à l'endroit adéquat, un fichier qui aura donc pour nom POA\_GRx\_NOM1\_NOM2.zip
- Votre répertoire, une fois décompressé, doit permettre l'exécution de votre solution : ne mettez pas le path de vos fichiers en absolu mais en relatif.
- le non respect de ces consignes d'envoi sera pénalisé.
- **Un programme ne compilant pas ou ne s'exécutant donne la note de 0 à ce travail.**
- **NB : Un plagiat donne la note 0 au module**
- vous serez interrogés sur ce travail individuellement au cours de la dernière séance. Des informations vous seront données ultérieurement.