

# Systeme de cotation de SVG Layout Designer

Par TECHFORM, Eric NGUYEN

- **Comment fonctionne le système de cotation de SVGLD ? (10 slides)**
- **Comment dessiner une cotation ? (5 slides)**
- **Quels sont les types de cotations ? (4 slides)**

# Comment fonctionne le système de cotation de SVGLD ?

# Comment fonctionne le système de cotation de SVGLD ?

## Rappel: Concept fondamental de SVGLD

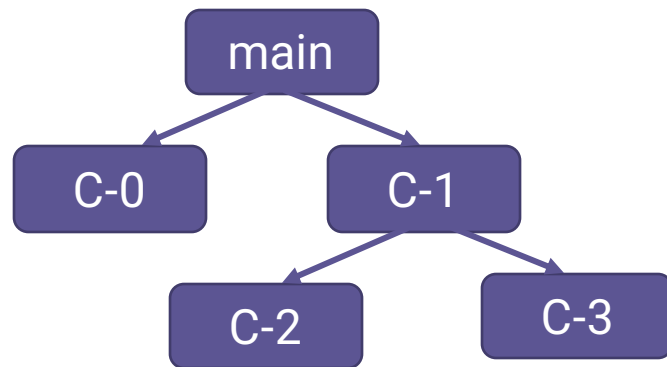
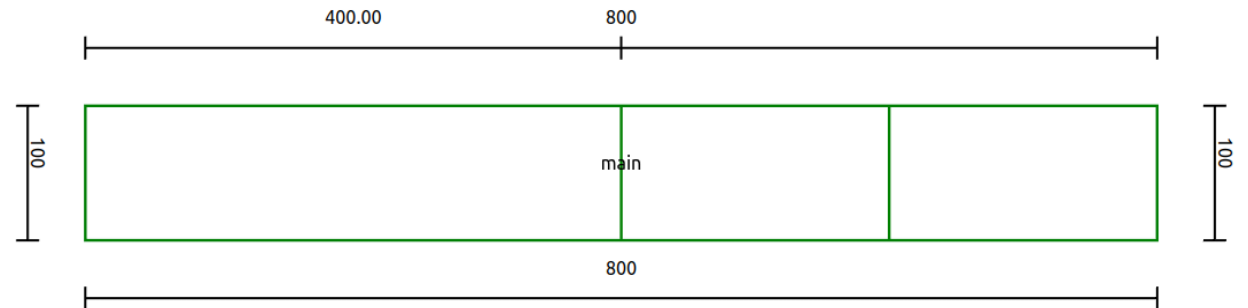
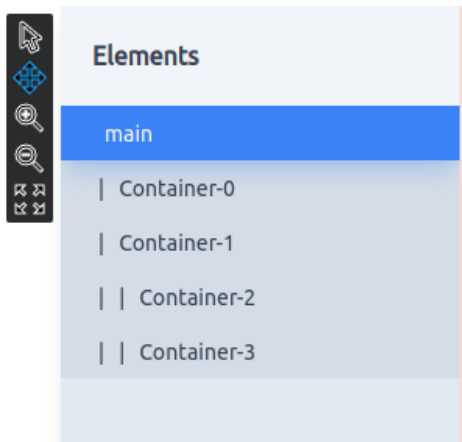
> SVGLD est un assembleur qui emboîte des conteneurs

- Il existe une hiérarchie entre les conteneurs
- En informatique, cette hiérarchie est appelée "arbre"
- Chaque conteneurs peut posséder une transformation horizontale (x) et/ou une transformation verticale (y)



# Comment fonctionne le système de cotation de SVGLD ?

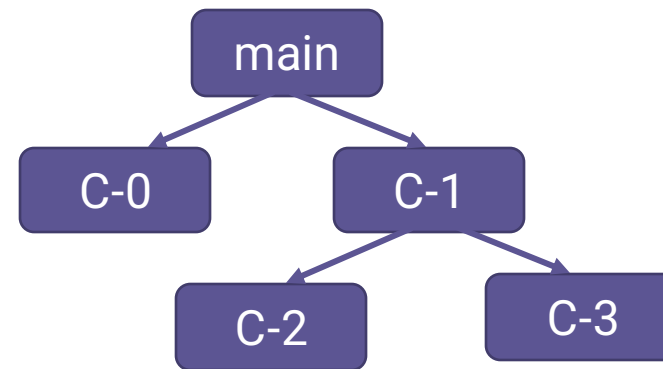
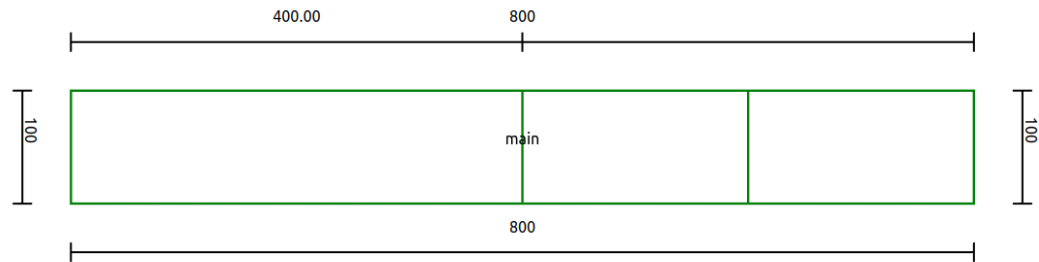
## Rappel: Concept fondamental de SVGLD (exemple)



# Comment fonctionne le système de cotation de SVGLD ?

## Rappel: Concept fondamental de SVGLD (exemple)

- main ne possède pas de transformation
- C-0 ne possède pas de transformation
- C-1 est décalé horizontalement de 400
- C-2 ne possède pas de transformation
- C-3 est décalé horizontalement de 200



# Comment fonctionne le système de cotation de SVGLD ?

Comment restaurer ces transformations pour nos côtes sans utiliser la hiérarchie ?

Il existe deux méthodes :

- 1) Itérer sur les parents du conteneurs parent d'un conteneur donné et faire l'opération inverse (addition)
- 2) Itérer sur tous les conteneurs et conserver la transformation

# Comment fonctionne le système de cotation de SVGLD ?

## Comment restaurer ces transformations pour nos côtes sans utiliser la hiérarchie ?

- 1) Itérer sur les parents du conteneurs parent d'un conteneur donné et faire l'opération inverse (addition)
- 2) Itérer sur tous les conteneurs et conserver la transformation actuelle

	Avantages	Désavantages
Méthode 1: itérer sur ses parents	<ul style="list-style-type: none"><li>- Simple</li><li>- peu couteux pour un seul appel</li></ul>	Coût $O(i)$ à chaque appel tel que $i$ est le niveau de la hiérarchie (qui est usuellement dans l'ordre de la dizaine)
Méthode 2: sauvegarder la transformation	<ul style="list-style-type: none"><li>- Se base sur un parcours d'arbre</li><li>- flexible</li></ul>	Coût $O(n)$ à l'appel tel que $n$ est le nombre de conteneurs (qui peut aller dans les centaines)





# Comment fonctionne le système de cotation de SVGLD ?

Comment restaurer ces transformations pour nos côtes sans utiliser la hiérarchie ?

Au premier coup d'oeil méthode 1 semble favorable.

Mais en réalité, méthode 2 est le moins couteux de tous par sa flexibilité pour résoudre notre problème.



# Comment fonctionne le système de cotation de SVGLD ?

## Quelle sont les difficultés du système de côtes ?

- Il existe plusieurs types de côtes...

=> ...Certaines dépendent de d'autres et requiert une vue globale sur tous les conteneurs (enfants ou parents)

=> Donc la méthode 1 nécessite en plus d'appliquer les méthode 2 sur les enfants pour obtenir leurs transformations

- Il y a beaucoup de conteneurs pour beaucoup de types de cotation...

=> ...on voudrait donc que toutes les côtes soient dessinées dans une même itération

=> au pire cas de la méthode 1, si on devait annuler la transformation pour tous les conteneur, le coût serait de l'ordre de  $O(n + k) > O(n)$

# Comment fonctionne le système de cotation de SVGLD ?

## Résumé

- On itère sur tous les conteneurs par parcours d'arbre
- On conserve la transformation courante
- On remonte les position restaurés des enfants dans l'itération

# Comment dessiner une cotation ?

# Comment dessiner une cotation ?

## C'est quoi une côte ?

Simplement dit, une côte est une *ligne avec deux lignes perpendiculaires*. Donc définir une côte revient à définir une ligne.

Une ligne a plusieurs méthodes de représentation en informatique/mathématique :

- 1) deux points d'extrémités (utilisé en SVG)
- 2) un vecteur, une longueur et une origine (utilisé pour les fonctions paramétrique)
- 3) un angle, un rayon et une origine (identique qu'au dessus)

# Comment dessiner une cotation ?

## C'est quoi une côte ?

Nous allons utiliser la méthode 1 pour la structure de donnée car c'est plus intuitif pour le paramétrage et pour le dessin de la ligne.

Mais nous allons également utiliser la méthode 2 (et 3), pour le dessin des lignes perpendiculaires.



Comment dessiner une cotation ?

**C'est quoi une côte ?**

La méthode 1 est supportée nativement par SVG (ou Canvas si c'est-ce que nous utilisons).

Expliquons donc la seconde méthode pour le dessin des lignes perpendiculaires.



# Comment dessiner une cotation ?

## Comment dessiner une perpendiculaire à une ligne ?

On cherche le vecteur perpendiculaire à notre ligne principale.

On commence par chercher le vecteur unitaire:

$$\Delta x = x_2 - x_1$$

$$\Delta y = y_2 - y_1$$

On obtient le vecteur unitaire en annulant la norme :

Norme =  $\sqrt{\Delta x^2 + \Delta y^2}$  par Pythagore

Vecteur normalisé =  $[\Delta x / \text{norme}, \Delta y / \text{norme}] = [\text{unitX}, \text{unitY}]$

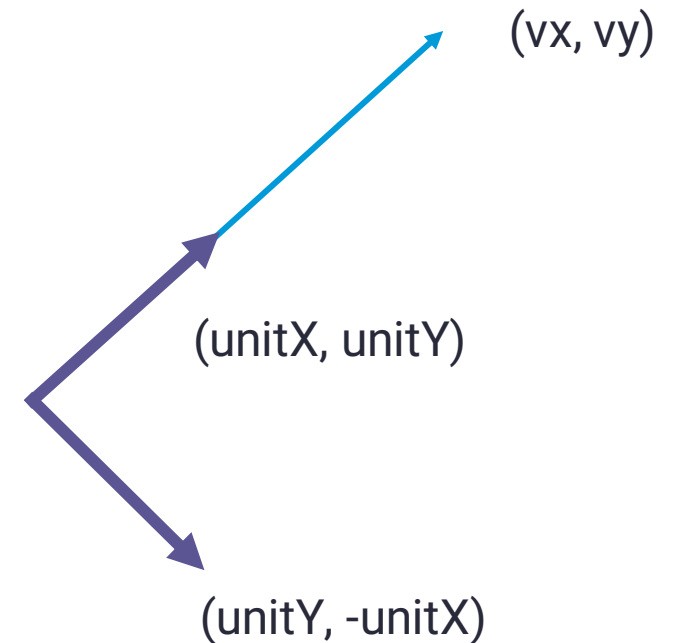
On obtient le vecteur perpendiculaire en faisant une transformation :

Vecteur perpendiculaire =  $[\text{unitY}, -\text{unitX}]$

PS: le texte utilise les deltas pour sa rotation en css par unité de tour:

Angle =  $\tan^{-1}(\Delta y / \Delta x) / 2\pi$  (en unité de tour,  $2\pi = 1$  tour et  $\pi = \frac{1}{2}$  tour)

(un peu couteux comme opération)





# Comment dessiner une cotation ?

## Comment dessiner une perpendiculaire à une ligne ?

Enfin on obtient les deux points d'extrémités de la perpendiculaire en utilisant la fonction paramétrique :

$$x = x_0 + t * v.x$$

$$y = y_0 + t * v.y$$

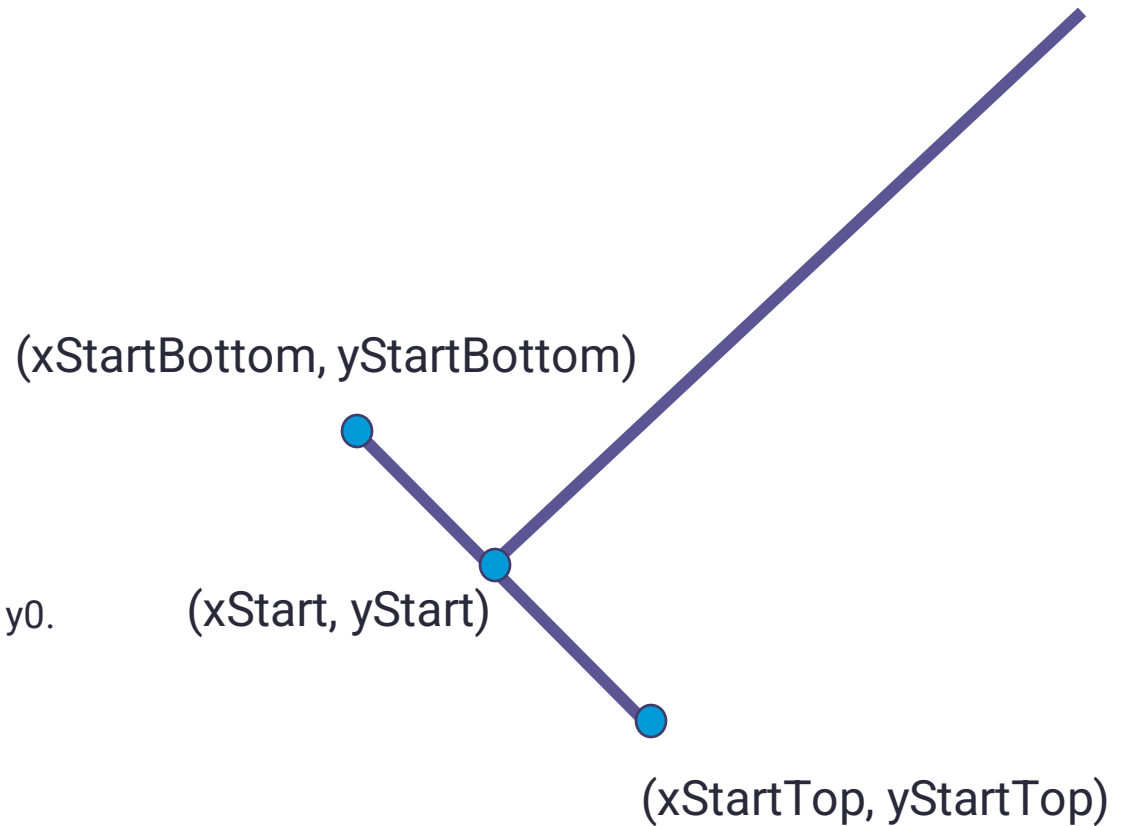
$$x_{StartTop} = x_{Start} + tailleLigne * v.x$$

$$y_{StartTop} = y_{Start} + tailleLigne * v.y$$

$$x_{StartBottom} = x_{Start} - tailleLigne * v.x$$

$$y_{StartBottom} = y_{Start} - tailleLigne * v.y$$

Pour la seconde extrémité de la côte,  
on peut réutiliser le même vecteur et juste changer  $x_0, y_0$ .



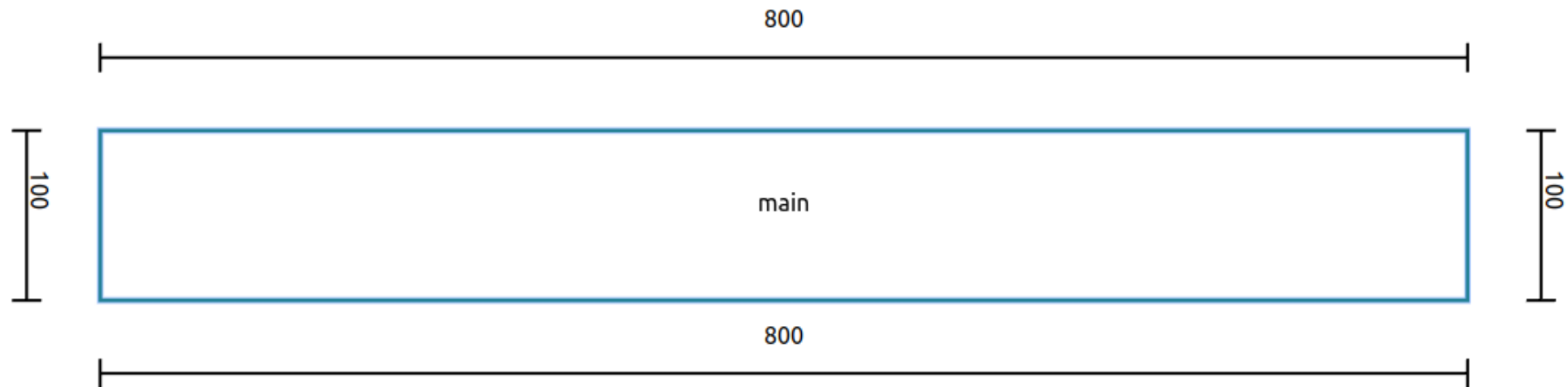
# Quels sont les types de cotations ?

# Quels sont les types de cotations ?

## Cotation propre (self-dimension)

La cotation propre est la dimension réelle d'un conteneur.

- Il ne depend que de lui-même et des transformations parents.
- En fonction d'orientation (horizontal/vertical) on prends la largeur ou la hauteur.

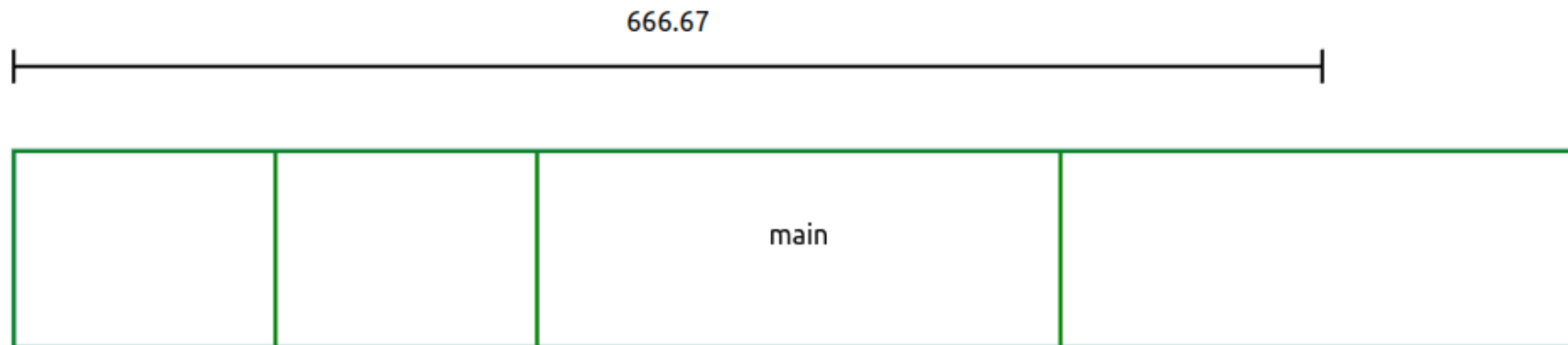


# Quels sont les types de cotations ?

## Cotation des enfants (children dimension)

La cotation des enfants est la cotation qui englobe la totalité des enfants.

- Il dépend de l'alignement et de la position des enfants (pas de ses sous-enfants par contre)
- Il calcule les min et max de ses enfants
- Il calcule x ou y en fonction de son orientation

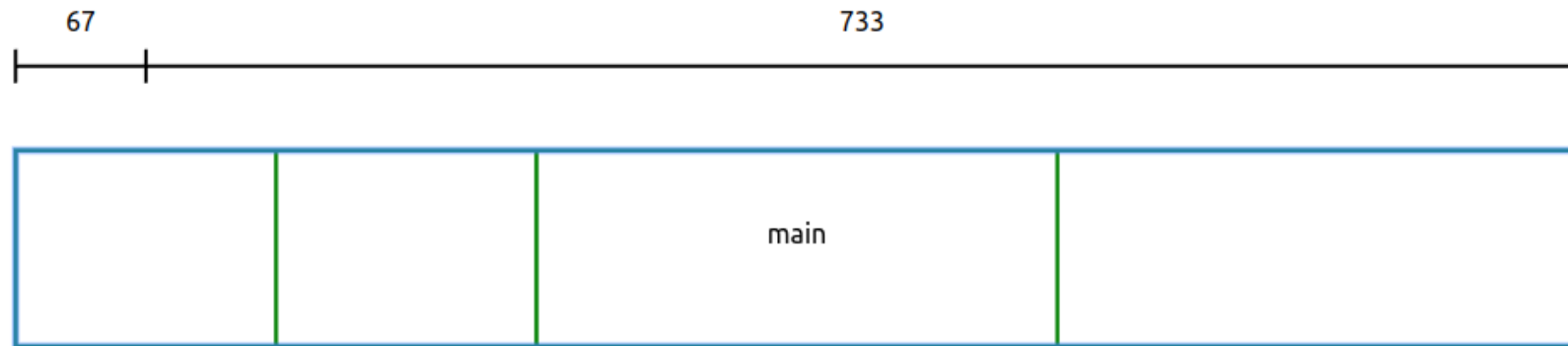


# Quels sont les types de cotations ?

## Cotation marquée des enfants (borrower dimension/dimension with marks)

La cotation emprunteur des enfants est la cotation qui englobe la totalité des enfants.

- Utilise pour extrémités ses propres dimensions
- En itérant sur ses enfants, il stocke les enfants "marqués" dans une liste avec leur position (avec leur alignement)
- Et dessine un trait vertical sur chaque position marquée (avec transformation restaurée)



# Quels sont les types de cotations ?

## Cotation par profondeur (depth dimension) (déprécié par children dimension)

La cotation par profondeur englobe tous les conteneurs dans une profondeur donnée.

- Utilise le parcours d'arbre en largeur
- Calcul le min et max

