

Programmation de techniques d'interactions en utilisant SVG 2015

Introduction

Rappelons ici qu'un document SVG peut être vu comme un arbre de scène dont la racine est un canvas SVG. Ce graphe peut être incorporé directement dans un document HTML5.

Clonez le squelette de projet suivant via GIT:

https://github.com/AlexDmr/ricm5_2015.git

Le projet contient un simulateur d'évènement multitouch afin que vous puissiez plus facilement développer sur vos machines.

Installation

Vous aurez besoin de :

- NodeJS et NPM
- GULP (npm install -g gulp)

Dans le répertoire, installez les dépendances avec la commande

npm install

puis lancez la compilation gulp:

gulp

Principe des transformations

Chaque noeud d'une page web peut contenir un attribut de style **transform** qui encode la transformation géométrique appliqué à ce noeud **par rapport à son parent**. Pour chaque noeud, il est possible de connaître sa matrice de transformation via la fonction XXX fournit dans le code squelette.

Vous pouvez consulter le document suivant qui présente différentes matrices de transformations 2D exprimées en coordonnées homogènes.

<http://morpheo.inrialpes.fr/people/Boyer/Teaching/RICM/c3.pdf>

Abonnement aux événements

Pour s'abonner à des événements, on utilise la fonction **addEventListener** sur les éléments du graphe de scène. Lors de l'abonnement aux événements *touch*, n'oubliez pas de désactiver le comportement par défaut (`preventDefault`) notamment pour éviter les problèmes sur les appareils mobiles (téléphones, tablettes, ...).

Vous pouvez consulter les différents types d'événements types d'événements [souris](#) et de [touché](#).

Abonnez vous aux événements du touché et affichez les dans la console de développement javascript (à l'aide la méthode `log` de l'objet `console`).

Quelques transformations

En SVG, une matrice de transformation contient 6 coordonnées, notées de a à f et correspondant à :

| | | |
|---|---|---|
| a | c | e |
| b | d | f |
| 0 | 0 | 1 |

Pour rappel, l'inverse d'une transformation matricielle peut s'obtenir (lorsqu'elle existe) en calculant la matrice inverse de cette transformation (méthode `inverse` de l'objet `matrix`). Dans la suite de ce TP, nous considérerons en particulier les transformations de translation et de similitude (ou Rotozoom, soit translation + rotation + mise à l'échelle).

Implémentation d'une translation via un pointeur

Il s'agit ici de déplacer un élément du graphe SVG à l'aide d'un pointeur (souris, doigt, ...). L'élément en question doit rester fixe sous le curseur.

En réutilisant une partie du code vu en cours (http://iihm.imag.fr/demeure/enseignements/M2MIAGE_PW/Cours/), implémenter cette transformation en vous basant sur les déplacements de la souris. Essayer ensuite de déplacer l'étoile rouge et la forme verte, que constatez vous ?

Il va falloir modifier un peu votre code pour prendre en compte de façon systématique le graphe de scène. Rappelons que l'attribut *transform* des noeuds SVG code la transformation par rapport au parent et que `getCTM` donne la matrice de transformation courante d'un noeud par

rapport au canvas SVG. Nous allons donc devoir modifier l'attribut *transform* du noeud en cours de translation. Il convient donc d'exprimer points et matrices dans ce repère :

1. Le point du pointeur exprimé dans le repère du parent : Utilisez `getCoordinate_relative_to` pour vous placez des coordonnées-page aux coordonnées du canvas SVG. Puis, appliquez la matrice de transformation inverse du père du noeud en cours de translation à ce nouveau point. Vous obtenez alors les bonnes coordonnées (faites des dessins si ça n'est pas clair !). **Soit P_{parent} ce point.**
2. Le point du pointeur exprimé dans le repère du noeud : Utilisez `getCoordinate_relative_to` pour vous placez des coordonnées-page aux coordonnées du canvas SVG. Puis, appliquez la matrice de transformation inverse du noeud en cours de translation à ce nouveau point. Vous obtenez alors les bonnes coordonnées (faites des dessins si ça n'est pas clair!). **Soit P_{noeud} ce point.**
3. La matrice de transformation du père au noeud : Il suffit pour l'obtenir de multiplier la matrice inverse du père avec la matrice de transformation du noeud. **Soit M cette matrice.**

On a alors la relation suivante : **$M \times P_{noeud} = P_{parent}$**

D'une façon plus générale, à chaque fois que le pointeur est déplacé, on calcule ses coordonnées dans le repère parent du noeud (et on les note **P_{parent}'**), et on cherche la matrice **M'** telle que : **$M' \times P_{noeud} = P_{parent}'$** où seules les coordonnées e et f sont à déterminer (a, b, c et d sont les même que pour M). En effet, nous cherchons bien la matrice de transformation qui permet de maintenir fixe la position du pointeur par rapport au noeud en cours de translation ! Ainsi l'équation matricielle précédente permet d'établir un système de deux équations à 2 inconnues qu'il suffit de résoudre pour déterminer e, f et donc M' .

À chaque déplacement du point responsable de la translation, il faudra donc recalculer M' et affecter l'attribut **transform** du noeud en cours de translation :

`transform="matrix(M'.a, M'.b, M'.c, M'.d, M'.e, M'.f)"`

Implémentation d'une transformation de similitude/Rotozoom

L'idée est la même que pour la translation, mais il faut considérer 2 points au lieu d'un seul, le système d'équations devient :

- **$M' \times P_{noeud_1} = P_{parent}'_1$**
- **$M' \times P_{noeud_2} = P_{parent}'_2$**

soient 4 équations, avec M' qui est de la forme :

| | | |
|------------------|-------------------|----|
| $e.\cos(\alpha)$ | $-e.\sin(\alpha)$ | tx |
| $e.\sin(\alpha)$ | $e.\cos(\alpha)$ | ty |

| | | |
|---|---|---|
|) | | |
| 0 | 0 | 1 |

soient 4 inconnues ($e.\cos(\alpha)$, $e.\sin(\alpha)$, t_x et t_y).

Seules les équations pour le calcul du Rotozoom diffèrent, il convient de bien prendre en charge l'ajout et le retrait de pointeurs pour coder l'interaction suivante :

- Si j'appuie avec un pointeur, je fais une translation
- Si j'ajoute un second pointeur, je fais un Rotozoom
- L'ajout d'autres pointeurs est ignoré
- Si je retire un des deux pointeur, je repasse en mode translation avec celui restant

Les solutions aux équations de la translation et du Rotozoom sont disponibles ici :

<https://docs.google.com/document/d/1U0Z-WYQ9kea-E0JkHjmscpb7hW1aDC-4YI4sPouc88c/edit>

Références

- <https://developer.mozilla.org/fr/docs/Web/API/TouchEvent>
-