TP Multi-touches 2016

Installation de l'environnement de développement

Clonez le squelette de projet suivant via GIT:

> ***git clone https://github.com/AlexDmr/Multi-touches.git***

Pour ce TP, nous programmerons avec les technologie web, vous aurez besoin d'installer :

- Webstorm : un éditeur de code dédié aux technologie web. Vous pouvez obtenir une licence gratuite auprès de l'éditeur.
- NodeJS et NPM : téléchargez la version 6.x de NodeJS.
- Gulp : permet de compiler le code. Installez le via un terminal :
  npm install -g gulp

Une fois ces prérequis vérifiés, rendez vous dans le répertoire du projet et, via un terminal, tapez la commande ***npm install*** pour installer les bibiothèques utiles au projet.

Editez le code dans Webstorm et lancer la tâche gulp default, la compilation tournera alors en tâche de fond.

Interaction multi-touches sur une base de photos

On propose de coder des interactions multi-touches sur des photos rendu sur une page HTML. Vous aurez à coder deux interaction :

1) La translation à l'aide d'un doigt
2) La translation + rotation + étirement à l'aide de deux doigts

La vidéo ci après illustre ces interactions : https://www.youtube.com/watch?v=tG2yc2dp7F8

Pour coder ces interaction, nous allons devoir manipuler des matrices et les appliquer aux images et vidéos rendues sur la page HTML. 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

Les navigateur web mettent à disposition des structures de données pour manipuler des matrices et des point: SVGMatrix et SVGPoint

Une SVGMatrix contient 6 coordonnées, notées de a à f et correspondant à :

a	c	e
b	d	f
0	0	1

Dans la section suivante, nous allons voir comment coder la translation à l'aide du doigt et en particulier ce que cela implique en terme de manipulation de matrices.

Implémentation d'une translation via un pointeur

Il s'agit ici de déplacer un élément HTML (ex: une image ou une vidéo) à l'aide d'un pointeur (souris, doigt, …). L'élément en question doit rester fixe sous le pointeur. Pour cela, vous aurez besoin des données suivantes :

1. Les coordonnées du pointeur lors du premier contact, exprimées dans le repère de l'élément. On nommera **Pt_coord_élément** le point exprimant ces coordonnées. Notez bien que les coordonnées du pointeur sont exprimées relativement au repère du document HTML, il faudra donc les convertir pour les exprimer relativement au repère de l'élément (utilisez la matrice de transformation inverse de l'élément).
2. Les coordonnées courante (c'est à dire au cours de l'interaction) du pointeur exprimées dans le repère du document HTML. On nommera **Pt_coord_parent** le point exprimant ces coordonnées.
3. La matrice de transformation du noeud, qu'on nommera **M**.

On a alors la relation suivante, lors du premier contact :
M x Pt_coord_élément = Pt_coord_parent

Au cours de l'interaction, cette relation n'est plus vérifié (le pointeur se déplaçant en de nouvelle coordonnées qu'on nommera **Pt_coord_parent'**), il faut alors calculer la matrice M' telle que :
M' x Pt_coord_élément = Pt_coord_parent'

Dans le cas de la translation, les coordonnées a, b, c et d de la matrice M sont constante, seules varient les coordonnées e et f. Ainsi, en résolvant l'équation matricielle précédente, il est possible de calculer e et f (cf ce document)

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' x P_noeud_1 = P_parent'_1**
- **M' x P_noeud_2 = P_parent'_2**

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

e.cos(α)	-e.sin(α)	tx
e.sin(α)	e.cos(α)	ty
0	0	1

soient 4 inconnues (e.cos(α), e.sin(α), tx et ty).

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/1UOZ-WYQ9kea-E0JkHjmscpb7hW1aDC-4YI4sPouc88c/edit

Implémentation

Dans le squelette de code qui vous est fournit, il faut compléter les fichiers :

- ts/MT_interactions.ts : Contient le squelette de l'automate d'interaction. La fonction **action** d'une transition de l'automate renvoie un booléen. Si ce booléen vaut vrai, alors la transition est franchie, si il vaut faux, alors la transition n'est pas franchie. La fonction est appelée en fonction lorsque un des **eventTargets** lève l'événement **eventName**.
- ts/transfo.ts : Contient les fonction à compléter pour coder le calcul de la translation et du rotozoom.

Pour démarrer, représentez sur une feuille de papier l'automate d'interaction représenté dans ts/MT_interactions.ts.