

Possibilités d'interactions temps-réel sur mobile

Document de vulgarisation pour Mr. Verdon dans le cadre du projet NIFFF 2014.
Ce document fait aussi office de choix technologique et de bref état de l'art pour le TP.

Introduction

La technologie des réseaux et des appareils mobiles d'aujourd'hui permet de se servir d'un Smartphone pour interagir à distance avec un autre appareil. Le document présent se limite aux technologies permettant à un utilisateur d'interagir avec une installation vidéo depuis un appareil mobile.

Dans le cadre du NIFFF 2014, l'interaction avec l'installation doit être possible que dans un périmètre restreint et cette interaction doit être la plus simple possible pour l'utilisateur.

L'interaction modifierait visuellement la projection en direct, ce qui contraste avec le principe d'un festival de film où les participants ont le rôle de spectateurs et ne peuvent pas agir sur l'image, mais seulement l'interpréter subjectivement.

Au vu des contraintes ci-dessus, on s'intéressera ici à une interaction par l'intermédiaire du navigateur de l'appareil, lui-même considéré comme connecté à un réseau wifi local.

Les navigateurs sont des logiciels complexes car ils rassemblent une énorme quantité de fonctions très diverses et les mettent à disposition par l'intermédiaire d'une seule interface unifiée. Les navigateurs mobiles se basent sur le même fonctionnement et les fonctions qu'ils proposent sont naturellement indépendantes de l'appareil utilisé.

Ce qui est important pour la simplicité de l'installation car l'utilisateur n'aura pas besoin d'installer une application particulière pour interagir avec l'installation. En effet un navigateur comme Firefox offre les mêmes fonctions sur n'importe quel appareil, bien sûr en restant dans les limites des possibilités de celui-ci.

Interactions possibles

La technologie implémentée dans la majorité des navigateurs web sur mobiles d'aujourd'hui permet de développer des applications très semblables aux applications dites « natives », celles que l'on doit télécharger et installer. La conception d'une « WebApp » répond donc à un nombre de plus en plus limité de contraintes.

L'installation du NIFFF sera composée d'un serveur qui contrôle les projections et de « clients » qui seront les utilisateurs de la WebApp. L'échange d'informations entre ses deux acteurs peut être à sens unique ou bidirectionnel. Les informations qui sont envoyées par le serveur peuvent être des commandes, du texte, des images (flux vidéo) ou autres chose.

Comme la communication est bi-directionnel, le serveur peut renvoyer des informations à l'appareil en retour aux commandes envoyée par celui-ci et vice-versa.

Depuis un navigateur mobile on pourra détecter les différents gestes effectué sur l'écran. Ces gestes sont les suivants : Clique, Double clique, Glissement, Zoom et Rotation. Ces gestes pourront être envoyés vers le serveur pour modifier la projection.

On pourra également avoir accès au capteurs de l'appareil. Soit l'accéléromètre, le gyroscope et un éventuel magnétomètre. On se limitera à utiliser l'accéléromètre car la grande majorité des Smartphones en possède un. Ce capteur permet de détecter l'orientation de l'appareil ainsi que la vitesse du déplacement sur les 3 axes.

La compatibilité des éléments ci-dessus pour les différents appareils est résolue par l'utilisation d'un navigateur qui fournit une interface unifiée pour l'accès au capteurs.

L'installation doit être très réactive et donc les communications réseau rapides c'est pour cela que une réseau wifi sera présent pour réduire la latence au maximum.

Interaction envisagée

Pour le prototype de base on enverra au minimum deux informations simples, à savoir la position du clique / tape de l'utilisateur dans le navigateur. La communication sera uni-directionnelle, le serveur n'envoyant rien au client.

Pour une installation finale, le mécanisme de l'interaction doit retenir l'attention du spectateur et provoquer de la curiosité. Pour obtenir cet effet il a été envisagé de mettre en place un jeu dans lequel chaque participants ou des groupes de participants seraient en concurrence les uns avec les autres.

On pourra différencier les différents joueurs ou équipes aux alentours de l'installation grâce à une couleur prédominante propre à chaque joueurs ou équipes, affichée sur chaque appareils client. Cette couleur correspondra à un élément projeté appartenant à une équipe ou un joueur.

On pourra également mettre en place un retour visuel animé sur le mobile. Par exemple une copie simplifiée de la projection ou transformer le mobile en manette de jeu tactile.

Le scénario précis du jeu est encore à définir, notamment le fait que les participants soient seuls ou doivent former des équipes pour jouer. Les règles et les limites précises ne sont pas encore établies. L'idée des équipes est attrayante car elle forcerait les participants à créer des connexions émotionnelles et le fait de pouvoir forcer les gens à faire des mouvements grâce au capteur de mouvement est intéressant.

Technologies

VISUALISATIONS

Les navigateurs mobiles donnent accès depuis longtemps à l'élément Canvas qui permet de rendre des éléments graphiques, à la manière d'une zone de dessin. Ce qui permet d'avoir une visualisation animée sur le mobile des clients.

Possibilité donc de partir « From Scratch » et dessiner dans l'élément Canvas. Néanmoins l'utilisation d'une librairie graphique est utile car elle peut fournir des Fallbacks et un plus haut niveau d'abstraction.

Les bibliothèques javascript les plus populaires sont D3.js¹ et Raphaël², on notera aussi processing.js³ qui a le désavantage de ne pas détecter les entrées de l'utilisateur (position du doigt, etc..) mais qui a l'avantage de se baser sur du code Processing et donc permet de réutiliser du code le cas échéant.

RÉSEAU

La majorité des navigateurs mobiles actuels ont mis en place le standard WebSocket, normalisé récemment par l'IETF dans la RFC 6455. A noter que le navigateur de base du système Android ne propose pas le support avant la version 4.4⁴, elle-même encore peu répandue.

WebSocket a l'avantage de travailler par le protocole HTTP tout en étant très performant une fois la communication établie sur un canal TCP (Max. 16 octets en plus des données du message transféré).

¹ "Data Driven Documents", <http://d3js.org/>

² <http://dmitrybaranovskiy.github.io/raphael/>

³ <http://processingjs.org/>

⁴ Le support est à vérifier, informations croisées avec <http://caniuse.com/#feat=websockets>

On envisagera un fallback vers un système du type AJAX suivant le résultat des tests de performances. En utilisant par exemple la librairie Socket.io⁵.

Choix technologique

Les choix technologiques suivant concernent uniquement l'application prototype et peuvent changer pour l'application finale.

Application serveur en Processing, offrant un serveur WebSocket⁶ qui reçoit les données des clients.

Application cliente simple, « WebApp » sur mobile, qui détecte la position du curseur ou du doigt et l'envoie au serveur, tout ceci peut être fait en javascript avec les fonctions fournies par les navigateurs (Touch Event, WebSocket, Sensors).

Références

« 10 Realtime Web Technology Predictions for 2014 » , <http://www.leggetter.co.uk/2014/02/24/10-realtime-web-technology-predictions-for-2014.html>

« High Performance Browser Networking », O'Reilly, Ilya Grigorik, 2013.
Lecture online : <http://chimera.labs.oreilly.com/books/1230000000545>

« L'art du game design », Pearson Education France, Jesse Schell, 2010.

« Mobile HTML5 Compatibility on iPhone, Android, Windows Phone and others mobiles devices », <http://www.mobilehtml5.org>

⁵ <http://socket.io/>

⁶ <http://java-websocket.org/>