

TP : GyroTouch Mouse

Objectifs

- se familiariser avec le pointage absolu et relatif
- se familiariser avec le filtrage pour les systèmes interactifs
- mettre en oeuvre des fonctions de transfert pour le pointage
- Manipuler des machines à état

L'objectif de ce TP est d'utiliser votre smartphone pour pointer sur un autre écran. Pour cela vous allez mettre en place plusieurs techniques de pointage. La première technique utilise l'orientation de votre smartphone pour contrôler la position absolue du pointeur à l'écran. La seconde technique utilise l'écran du smartphone pour contrôler le pointeur à l'écran de manière relative, de façon similaire à un touchpad. Enfin la troisième technique combine des deux premières.

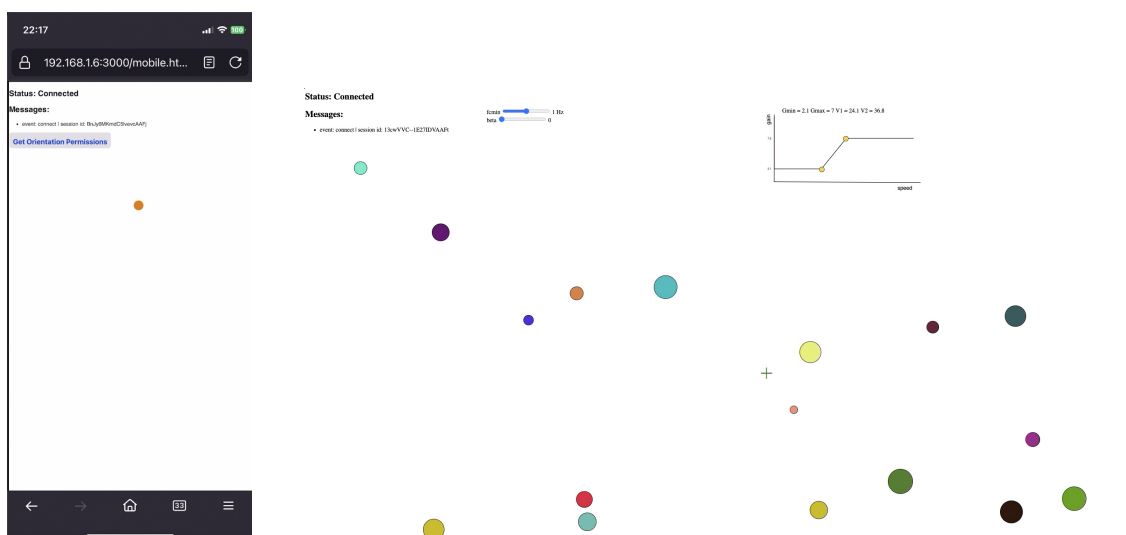


FIGURE 1 – l'interface du client sur le mobile à gauche et interface de l'ordinateur de bureau à droite.

1 Pré-requis

1. Suivez les instructions disponibles sur l'intranet <https://intranet.fil.univ-lille.fr/2020/04/09/nodejs-et-npm/> pour installer la dernière version de node et npm sous Linux.
2. Configurez le VPN ULille sur votre smartphone en suivant ces instructions <https://infotuto.univ-lille.fr/fiche/VPN>

2 Matériel à disposition

Après avoir décompressé les sources du TP, allez dans le répertoire `code/server` et installez les packages node nécessaires avec la commande `npm install` puis lancer le serveur avec la commande `npm run start`.

Faites la même chose avec les clients dans le répertoire `code/clients` : utilisez la commande `npm install` puis `npm run watch`.

Si tout s'est bien passé, vous pouvez vous connecter depuis votre smartphone à l'adresse `https://b02pXX.fil.univ-lille.fr:3000/mobile.html` où vous remplacerez `XX` par votre numéro de machine. Bien sûr, pas besoin de VPN si vous êtes chez vous (remplacez simplement `b02pXX.fil.univ-lille.fr` par l'IP de votre machine avec le serveur). Sur la machine de bureau, ouvrez également l'url `https://localhost:3000/desktop.html`.

Pour la suite du TP les modifications de code sont à réaliser dans le fichier `desktop.js`, sauf mention contraire.

3 Pointage absolu

Q 1. Les événements d'orientation sont reçus dans la méthode `socket.on("orientation"...`. Déplacer le curseur avec les mouvements d'orientation du smartphone et ajustez la valeur de gain pour atteindre tout l'écran avec des rotations confortables du poignet. Quelle valeur avez-vous choisi ?

Q 2. Suivant le modèle de votre téléphone, la position du pointeur peut être plus ou moins bruitée. Si votre pointeur est particulièrement stable, ajouter du bruit en utilisant la méthode `gaussianRandom`. En utilisant les sliders disponibles sur l'interface, ajustez les paramètres du `1€ filter`¹ pour réduire ce bruit sans ajouter trop de latence. Quel est le modèle de votre smartphone ? Comment avez-vous procédé pour régler les paramètres ? Quelles valeurs obtenez-vous ? Note : `xnoisy` et `ynoisy` peuvent être utilisés pour montrer la différence entre avant et après le filtrage.

Q 3. En utilisant la librairie `javascript-state-machine`², créez une machine à états qui permet de déclencher une sélection (`click()`) après un touch up. Note : vous pouvez exporter la visualisation de votre machine à états avec la librairie³. Concrètement, vous pouvez remplacer `visualize(fsm)` par `console.log(visualize(fsm))` pour afficher le code du graphe en syntaxe dot dans la console. Faites ensuite un copier-coller dans un fichier. Graphviz vous permet ensuite de convertir le fichier dot en SVG (par exemple) avec la commande `dot -Tsvg test.dot > test.svg`.

Q 4. Vous allez maintenant remplacer le téléphone par l'IMU du TP sur le prototypage hardware⁴. Vous allez dans un premier temps réaliser une fusion des données de l'accéléromètre et du gyroscope pour obtenir l'orientation de l'IMU, en utilisant la librairie `SensorFusion`⁵ et l'algorithme de Mahony⁶ (aidez-vous de l'exemple fourni avec la librairie⁷). Vous enverrez les informations d'orientation en écrivant sur le port série le pitch, roll et yaw, séparés par des virgules. Pour tester, décommentez le code à fin du fichier `server.js` et modifiez le port série en conséquence. Pour le compte-rendu, vous fournirez le code Arduino qui permet de répondre à la question (fichier ino).

Pour la suite du TP, vous utiliserez seulement votre téléphone.

4 Pointage relatif

Q 5. Modifiez votre machine à états pour réaliser un pointage relatif avec l'écran du smartphone et mettez à jour les fonctions `touchstart`, `touchmove`, `touchend` et `orientation`.

Q 6. Vous allez maintenant mettre en place une fonction de transfert pour minimiser le débrayage du doigt lors du parcours de longues distances sur l'écran. Modifiez la fonction `touchmove` pour utiliser la fonction `sigmoid` et ajustez ses paramètres avec l'interface. Quelles valeurs de `Gmin`, `Gmax`, `V1` et `V2` avez-vous obtenu pour parcourir facilement la largeur de l'écran et rester précis ?

1. <https://gery.casiez.net/leuro/>

2. <https://github.com/jakesgordon/javascript-state-machine>

3. <https://github.com/jakesgordon/javascript-state-machine/blob/master/docs/visualization.md>

4. <https://thomaspietrzak.com/teaching/IHMA/tparduino.html>

5. <https://github.com/aster94/SensorFusion>

6. <https://doi.org/10.1109/TAC.2008.923738>

7. <https://github.com/aster94/SensorFusion/blob/master/examples/SensorFusion/SensorFusion.ino>

5 Combinaison des pointages absolu et relatif

Q 7. Modifiez une dernière fois votre machine à états pour combiner les pointages absolu et relatif : le pointage relatif est utilisé quand l'utilisateur déplace son doigt sur l'écran du smartphone et le pointage absolu est utilisé sinon. Le passage de pointage absolu à pointage relatif est immédiat quand l'utilisateur touche l'écran. Par contre, le passage pointage de relatif à absolu se fait au bout de 1000 ms par défaut, afin de permettre à l'utilisateur de débrayer en relatif. Cette technique de pointage est inspirée de la technique Laser+Track de Nancel et al., page 19, figure 10.

6 Compensation d'une partie de la latence avec le TurboTouch predictor

Q 8. Après avoir installé la librairie `turbotouchpredictor`, testez-le en affichant le curseur prédit en plus des autres. Testez également différentes valeurs de compensation. Quelle valeur de compensation fonctionne le mieux ? Pour plus d'informations sur le TurboTouch predictor, consultez la page du projet.