IHM Projet
Tremor : le tapis musical

Antoine Friant
Valentin Finini
Lawrence Stalder

21 janvier 2018

1 Introduction

Ce projet consiste à concevoir et implémenter une interface homme-machine à partir de capteurs de vibration Phidget.

Tremor ("secousse") est un application qui fonctionne avec ces capteurs de vibration au sol et permet de choisir un son qui se produit à l'activation d'un capteur. Si les capteurs sont fiables et bien calibrés, il devrait être possible de faire de la musique en tapant des pieds !

2 Guide d'installation

2.1 Pilote

Suivez le guide d'installation du pilote Phidgets (ignorez la section Phidget Network Server) :

- Sur Linux : https ://www.phidgets.com/docs/OS_-_Linux
- Sur Windows : https ://www.phidgets.com/docs/OS_-_Windows
- Sur macOS : https ://www.phidgets.com/docs/OS_-_macOS

Sur une machine Linux, il est nécessaire de suivre la section "Setting udev Rules" en bas de page afin de faire fonctionner l'application sans avoir besoin des privilèges administrateur.

Java 8 doit être installé pour lancer Tremor.

2.2 Matériel

- Munissez vous d'un PhidgetInterfaceKit (phidget ID : 1018_X) ainsi que de 1 à 8 Vibration Sensors (phidget ID : 1104_0).
- Connectez les capteurs aux entrées analogiques du kit d'interface. La position d'un capteur sur le kit d'interface déterminera son numéro dans l'application : de gauche à droite, ils seront numérotés de 0 à 7.

- Connectez le kit d'interface à l'ordinateur *via* USB.
- Posez les capteurs au sol. Ils devraient être espacés d'au moins 1 mètre et les disques métalliques ne doivent pas être en contact avec le sol (utilisez le poids d'un petit objet, ou du ruban adhésif pour les fixer).

3 Guide d'utilisation

Pour lancer Tremor double cliquez sur l'archive Tremor.jar ou, si votre système n'est pas configuré pour exécuter des archives jar, ouvrez un terminal à l'endroit où se trouve Tremor.jar et exécutez la commande "`java -jar Tremor.jar`". La fenêtre principale dont un exemple est illustré en figure 1 apparaît alors.

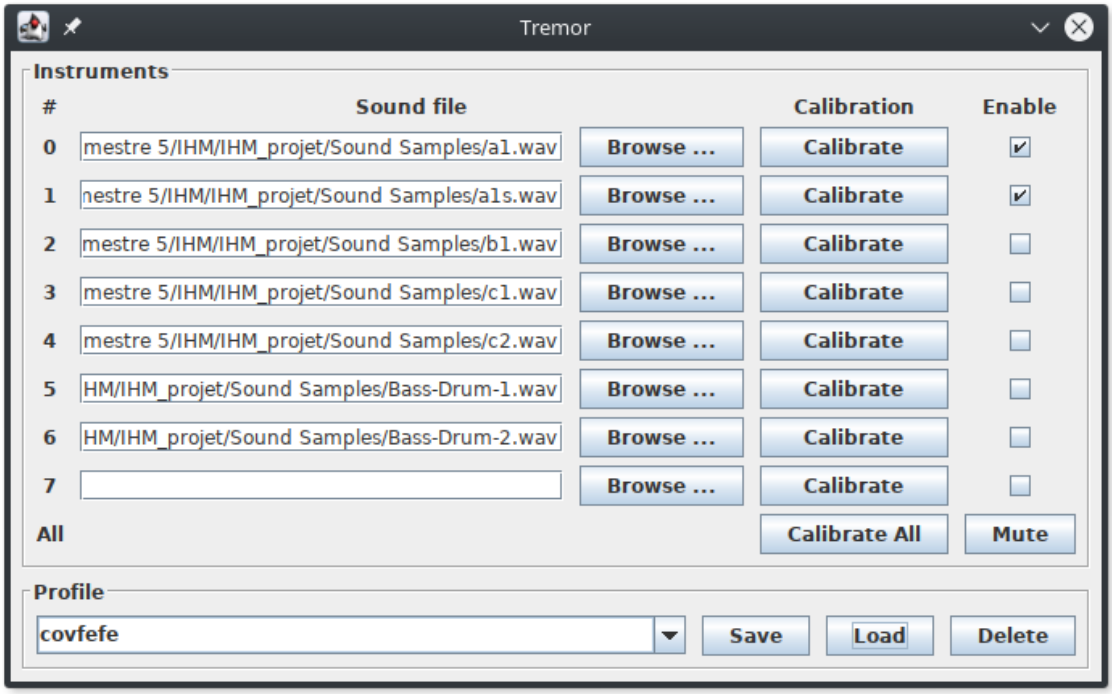

FIGURE 1 – Fenêtre principale

Pour chaque instrument correspondant à un capteur (numérotés de 0 à 7) on peut sélectionner un fichier audio qui sera lu lorsque l'utilisateur tapera du pied à côté du capteur. Les formats audio supportés par Tremor sont les mêmes que la bibliothèque standard Java : AIFF, AU et WAV.

Il est fortement recommandé de calibrer les capteurs ("Calibrate All" fait toutes les calibrations d'un coup). Pour cela, cliquez sur le bouton "Calibrate", puis tapez des pieds au sol pendant 5 secondes (un timer s'affiche). Pour obtenir les meilleurs résultats, utilisez "Calibrate All" et tapez quelques coups sur un point équidistant à tous les capteurs.

Il est possible de sauvegarder un profil (contenant fichiers, calibration et instrument activé ou non). Pour cela, il entrez un nom de profil et cliquez sur "Save". S'il existe déjà, une confirmation sera demandée. Il apparait alors dans la liste déroulante et peut être chargé en cliquant sur "Load", ou supprimé avec "Delete".

Si l'application se comporte anormalement (crash ou capteurs non détectés), essayez de la relancer avec les privilèges administrateur.

4 Problèmes rencontrés

4.1 Java et ALSA

Ayant développé l'application en Java sur Linux, nous avons fait face à une difficulté inattendue : Java et le pilote ALSA ne s'aiment pas. Bien que beaucoup de problèmes de compatibilité furent résolus au fil des années, nous en avec rencontré un autre : Java lit les fichiers audio sur ce qu'il considère comme la carte son par défaut, plutôt que la carte son en cours d'utilisation. Il fallu donc explicitement configurer ALSA pour que la sortie par défaut soit le haut-parleur plutôt que la prise HDMI sur laquelle rien n'était connecté.

Malgré cela, il se peut que Java lève une erreur signalant que le canal audio est occupé. Redémarrer l'ordinateur règle ce problème.

4.2 Calibration

4.2.1 Seuil d'activation

Nous nous sommes trop hâtivement lancés dans la programmation sans prendre le temps de lire la documentation des capteurs ou comprendre les données qu'ils émettent. Les capteurs n'émettent en réalité pas un voltage correspondant à l'état de l'onde de choc à l'instant de la lecture des données. Ils effectuent un échantillonnage toutes les 16 ms (fréquence maximale) et ne renvoient que le voltage moyen du dernier échantillonnage. Il est donc impossible de lire de façon directe l'onde de choc comme on pourrait la lire sur un oscilloscope.

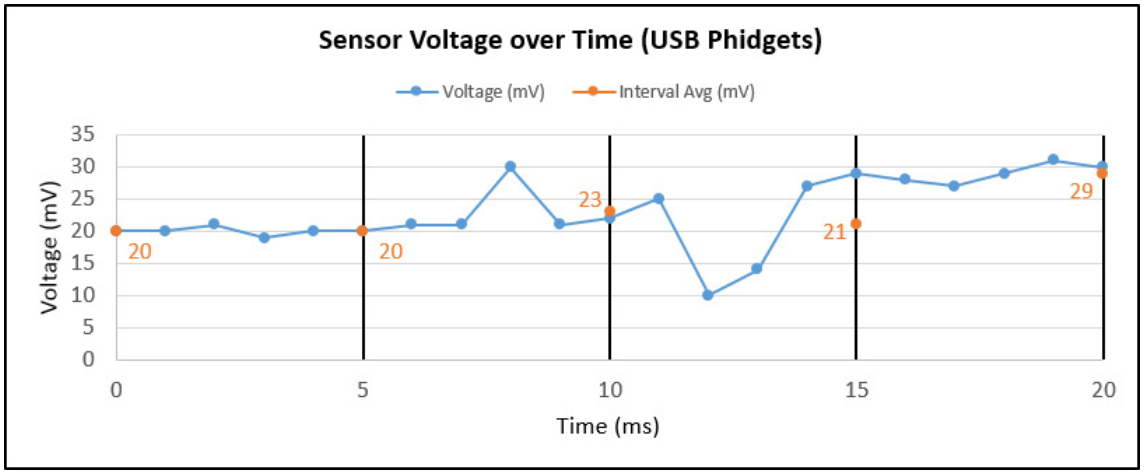

FIGURE 2 – Exemple fictif d'échantillonnage (5 ms) (source : *phidgets.com*)

En observant les données reçues des capteurs, nous avons remarqué qu'il détectaient des interférences encore inexpliquées provocant des pics de voltage mais pas de chutes. Lorsqu'on tape du pied, le voltage descend puis monte. Comme les diminutions soudaines de voltage n'arrivent que lorsqu'on tape au sol, nous avons décidé de baser la détection de vibrations sur les chutes soudaines de voltage.

Lors de la calibration qui dure 5 secondes, la valeur mesurée la plus basse est considérée comme le seuil d'activation. Un instrument joue un son lorsque la valeur reçue de son capteur descend en-dessous du seuil d'activation.

4.2.2 Activations multiples

Un problème supplémentaire était encore à régler : lorsqu'on tape au sol, tous les capteurs reçoivent la vibration à un intensité très similaire. Afin d'éviter que tous les instruments jouent un son en même temps à chaque input, nous avons pensé à n'activer que l'instrument donc le capteur s'active en premier. Ce n'est pas possible car la vitesse de l'onde de choc dans le bois est de 1 à 4 m/ms, soit 20 fois trop rapide pour l'intervalle d'échantillonnage de nos capteurs.

Nous avons donc opté pour l'activation de l'instrument dont le capteur est le plus en-dessous du seuil d'activation. Un fois qu'un instrument est activé, aucun instrument ne peut être réactivé pendant 50 ms. Malgré cela, il arrive encore que le mauvais instrument soit activé à cause du manque de précision des capteurs et surtout, d'après nous, la perte de précision due à l'échantillonnage effectué par les capteurs.