# BCI-Music Architecture

Le projet vise à connecter différents *wearables* (principalement des BCI) à une application interactive musicale. Le développement sera fait sous PureData qui est nonseulement multiplateforme, mais qui permet aussi l’interconnexion entre les dispositifs et le contenu musical, sans interfacer avec d’autres ressources (DAW, plug-in, application, etc.).

L’ordinateur traitant le contenu musical sera appelé serveur. Les ordinateurs interfaçant avec les *wearables* seront appelés clients. Ces derniers communiquent les données via UDP à l’adresse IP du serveur.

Pour assurer une communication avec moins de latence, le réseau devra être filaire. Toutefois, si les résultats obtenus avec un réseau sans-fils sont satisfaisant, l’option pourra être considérée.

Instructions

Cette section est vouée à changer avec le développement des fonctionnalités.

1. Lancer PureData
2. Ouvrir client.pd
3. Ouvrir serveur.pd
4. Dans la fenêtre client, appuyer sur « connect localhost 13000 »
5. Dans la fenêtre client, appuyer sur le carré doté d’un cercle au milieu. (Dans PureData, cet objet est reconnu comme un bang)

Mapping Audio

|  |  |
| --- | --- |
| BCI 1 | Concentration : Intensité des loops de drums  Méditation : Textures ou bruits ambients |
| BCI 2 | Concentration : Sub-bass  Méditation : Textures ou bruits ambients |
| MYO 1 | Main : Sélection du son  Accéléromètre : Delay |
| MYO 2 | Accéléromètre : Filtre sur la performance |
| ECG | Rythme : bpm |

Tâches

Dans chacun des cas, la première étape est de déterminer la méthode de communication avec le dispositif (sérielle, Bluetooth, etc.).

La section ressources donne des indications pour les différentes approches avec PureData.

* Création d’un *patch* client\_muse.pd envoyant au serveur les données d’un casque **Muse** au serveur. Il faudra identifier les signaux disponibles plus clairement, la communication peut se faire via OSC, mais implique d’utiliser le logiciel MuseIO.
* Création d’un *patch* client\_openbci.pd envoyant au serveur les données d’un **OpenBCI** au serveur. La principale donnée à recueillir est celle du cœur (ECG), mais le dispositif permet aussi de capter le cerveau (EEG) et les muscles (EMG)
* Création d’un *patch* client\_myo.pd envoyant au serveur les données d’un bracelet **MYO** au serveur. Le bracelet traite 5 positions différentes pour la main et 3 axes avec l’accéléromètre
* Design audio et composition de loops.
* Création d’un scénario côté serveur pour traiter les données avec l’audio.

Ressources

* PureData (télécharger la version Pd-extended)   
  <https://puredata.info/downloads>
* Communication sérielle via l’objet *hid* de PureData  
  <http://en.flossmanuals.net/pure-data/sensors/game-controllers/>
* Communication udp et osc via *udpreceive* et *unpackOSC* de PureData  
  <http://en.flossmanuals.net/pure-data/network-data/osc/>
* Muse <https://sites.google.com/a/interaxon.ca/muse-developer-site/home>
* MYO  <https://developer.thalmic.com/docs/api_reference/platform/index.html>
* OpenBCI <http://docs.openbci.com/tutorials/01-GettingStarted>