Numéros / n° 4 - automne 2014

« La pédagogie face à l'informatique musicale »

Lionel Rascle

« Il faut choisir, se reposer ou être libre »
Thucydide

1. État des lieux

Lorsqu'il y a cinq ans j'ai voulu créer des classes d'informatique musicale dans les deux écoles de musique municipales dans lesquelles je travaille en tant qu'assistant territorial d'enseignement artistique, je me suis questionné sur ce qui pouvait être en jeu dans l'apprentissage de la musique par rapport aux logiciels eux-mêmes.

En quoi des applications pouvaient aider les élèves à apprendre mieux ? Comment des logiciels pouvaient-ils permettre de comprendre les différents enjeux présents dans l'écriture musicale ? Était-il possible, d'un point de vue plus technique, de faire appréhender d'une manière claire ce qu'est un message MIDI par rapport à un flux audio ? Comment voir par où passe le signal à partir de visualisations simples ?

Dans le cadre, par exemple, de l'édition de partition lors d'un cours de formation musicale, il était clair que l'utilisation de logiciels comme Finale, Sibelius, Musescore, GuitarPro ou TuxGuitar n'était pas envisageable. En effet, il s'agit, avec ceux-ci, de manier une interface graphique, mais pas forcément de comprendre les différents éléments du langage musical. Et puis on peut réellement se questionner sur la place d'un enseignant vis-à-vis de ce type de GUI (1) qui offre tout, mais ne questionne pas l'apprenant. Finalement, on se retrouve plus à faire de la mise en page, qu'à écrire réellement consciemment de la musique.

Même dans le cadre de la musique assistée par ordinateur, il n'est pas satisfaisant, pour une compréhension simple, d'utiliser de gros logiciels comme CubaseVst ou ProTools. Là encore on se perd dans les menus et dans les multiples possibilités que l'on a et on se retrouve vite en dehors du sujet qui est d'apprendre aux élèves par où passe le signal et ce que l'on peut en faire.

Souvent, ce type de cours fait sur ce type de logiciels n'est pas réellement pédagogique mais se borne à être une démonstration. On montre ce que l'on peut faire avec, mais on ne questionne pas l'apprenant sur ce qui fait le son et sa transformation. On est attiré par les multiples possibilités, et du coup l'utilisation de celles-ci prend le pas sur une réelle réflexion de ce que l'on veut produire en amont.

Prenons un exemple simple au niveau de l'édition musicale, avec l'éditeur MuseScore. Il s'agit d'un logiciel d'écriture musicale libre qui s'apparente à Finale ou Sibelius, c'est-à-dire que l'on met graphiquement les notes sur une portée. Lorsque l'on crée une nouvelle partition et que l'on arrive à la sélection de l'armature, on a en visuel les différentes possibilités de tonalité, et lorsque l'on passe la souris sur ces pictogrammes, le nom des tonalités apparaît... Tout est fait, rien n'est à dire, il n'y a pas de pédagogie possible ici. Une réflexion de la part des élèves par rapport aux tonalités est alors tronquée.

Plus grave, lorsque l'on veut changer les rythmes des notes que l'on veut insérer dans la partition, il y a la possibilité de le faire sur le pavé numérique, on a :

1 = quadruple croche

2 = triple croche

3 = double croche

4 = croche

5 = noir

6 = blanche

7 = ronde

Comment faire un lien pédagogique avec tout cela pour dire que la ronde a pour valeur 1, la blanche 2, la noire 4, la croche 8 et du coup faire des retours possibles sur des expressions comme 4/4, 6/8 ou 2/2 directement issues de ces valeurs ? C'est impossible.

2. Alternatives

Il faudrait donc un logiciel qui permette d'appréhender la pensée musicale et de questionner l'élève sur chaque signe que l'on place sur la partition. Un logiciel qui permette de penser la musique en l'écrivant, et non pas de faire seulement de la mise en page. Pour moi, Lilypond (2) est exactement ce dont j'ai besoin.

Cela dit son approche peut être assez déconcertante. Ce n'est pas un logiciel WYSIWYG (*What You See Is What You Get*), car il faut écrire la musique tout d'abord dans un fichier texte, puis compiler ce fichier avec Lilypond pour avoir la partition en PDF ou en MIDI. Historiquement, Lilypond vient lui-même de la notation ABC qui permet d'écrire un morceau de musique avec les caractères ASCII. Cela implique donc la connaissance du codage avec ce logiciel, codage qui décrit précisément tout ce qu'il y a dans la partition de façon très musicale. Mais ce logiciel va beaucoup plus loin, car il a été créé pour se rapprocher le plus possible des prérequis des maîtres graveurs du xix^{e} siècle. En effet, le côté très mécanique de ce qu'est capable de faire Finale par exemple, n'aide pas à la lecture, les concepteurs ont donc mis sous forme d'algorithmes tous ces prérequis.

Dans un cadre général un peu plus informatique, ce logiciel permet également de montrer que l'interface graphique n'est pas le logiciel, de comprendre comment fonctionne réellement un programme et de toucher du même coup à la programmation. Suite aux différents cours que j'ai effectués, je me suis rendu compte que les personnes qui avaient le plus de mal avec cette application étaient précisément ceux qui avaient l'habitude de travailler sous des logiciels WYSIWYG.

Ce qu'apporte en plus ce logiciel c'est le fait qu'il soit libre. Son code étant visible et modifiable, on peut le faire fonctionner sur Open Office ou Libre Office. À noter également qu'il existe une application Web (3) où l'on peut voir ce que l'on écrit sous forme de partition. Cela permet donc d'essayer le logiciel sans l'installer.

3. Quelques exemples

J'ai montré précédemment le problème qu'il y avait à créer des liens pédagogiques avec des logiciels

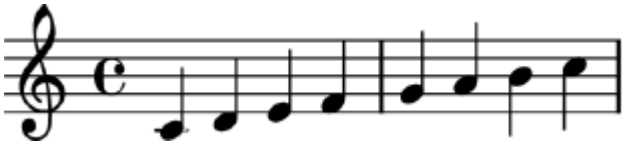
comme Musescore. Je vais maintenant décrire quelques exemples qui vont nous permettre de créer ces liens. Je ne vais pas ici refaire un cours complet sous Lilypond, vous pouvez pour cela vous référer au manuel d'initiation (4).

Mon intention est de montrer les allers et retours pédagogiques possibles.

3. 1. La hauteur et les rythmes

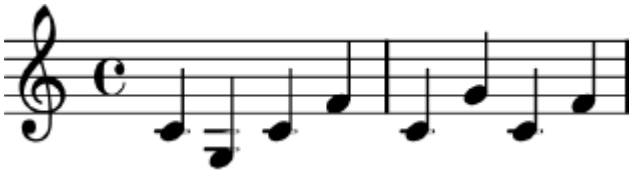
Avant toute chose, le code doit être mis entre accolades. Voici un premier exemple :

```
{relative { c d e f g a b c }}
```



relative permet que les hauteurs de notes soient relatives les unes aux autres jusqu'à l'intervalle de la quarte en partant du do grave. Au-delà de cet intervalle, on doit se servir de la virgule ou de l'apostrophe pour descendre ou monter la note d'une octave :

```
{relative { c g c f | c g' c, f}}
```



L'élève est donc obligé ici d'imaginer le type d'intervalle auquel il a à faire pour que l'exemple musical soit le même que celui qu'il a à écrire.

Les barres verticales symbolisent les changements de mesure. Grâce à celles-ci, Lilypond émet des messages de mise en garde si les rythmes écrits ne respectent pas la mesure, celle-ci étant mise à 4/4 par défaut. L'élève doit trouver par lui-même l'erreur et comprendre ce qu'il a écrit. Voici un exemple :

```
{relative { c d e f g | }}
```

Lilypond va essayer de générer ce que l'on a demandé d'écrire mais un message de mise en garde va apparaître :

```
Avertissement : échec de contrôle de mesure (barcheck) à : 1/4
```

```
{relative { c d e f g
                        | }}
```

Pour les rythmes, ceux-ci sont complètement en lien avec le système musical :

1 = ronde

2 = blanche

4 = noire

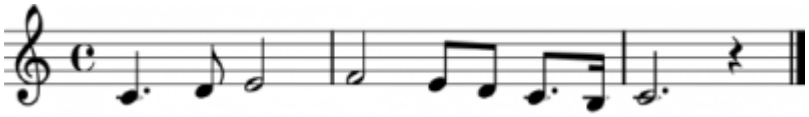
8 = croche

16 = double croche

etc.

Le point ajouté après le rythme aura le même effet que si l'on mettait une valeur pointée à une note. Voici un exemple avec des rythmes différents :

```
{relative{c4. d8 e2 | f e8 d c8. b16 | c2. r4 | bar "|."}}
```



Ainsi, comme je le disais précédemment, les allers et retours concernant la théorie musicale autour des rythmes et de la mesure sont tout à fait possibles ici. Lorsque l'on parle par exemple d'une mesure à 6/8, on peut faire référence explicitement à la croche et les élèves peuvent alors faire les liens indispensables à la compréhension de ce type de notation.

On remarque également que l'on peut rajouter des barres de mesures différentes grâce à l'outil **bar**.

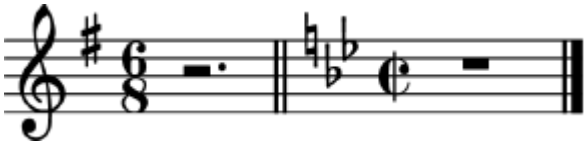
3. 2. Tonalités et clés

C'est l'outil **key** qui va permettre d'exprimer une tonalité. Il faut mettre ensuite le nom de cette tonalité ainsi que son type. Dès lors une tonalité de sol majeur s'écrira en mettant **g** puis **major.**

Voici quelques exemples :

```
{relative {

time 6/8

key g major

r2.

bar "||"

time 2/2

key g minor
```

```
r1

bar "|."

}}
```



On voit que chaque élément d'une partition est nommé, cela permet à l'élève de s'arrêter sur ces différents signes et d'y réfléchir. Le fait d'exprimer cela par un nouveau langage permet, à mon sens, d'aborder d'un point de vue différent l'écriture musicale. Un peu comme lorsqu'en pédagogie il faut essayer de faire comprendre une idée avec plusieurs entrées possibles.

4. Les logiciels libres au secours de la pédagogie ?

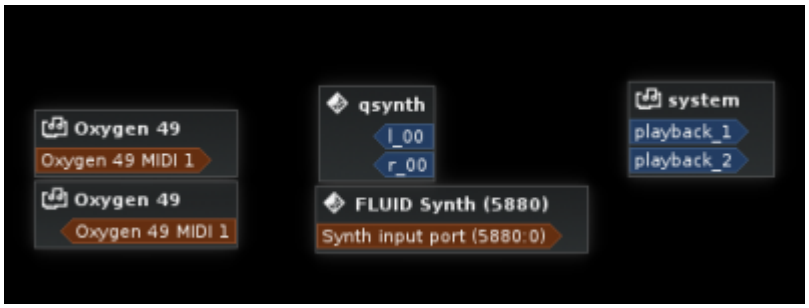
On a vu que Lilypond nous permettait de faire apparaître de façon explicite les différents éléments du langage musical et cela parce que Lilypond a été conçu uniquement pour écrire de la musique, et non pas comme un séquenceur, enregistreur. Certes Lilypond peut générer un fichier MIDI de ce que l'on écrit, mais il s'agit juste d'écouter de manière basique la partition.

Ce type de logiciel qui ne se borne à faire qu'une seule chose, mais d'une manière assez parfaite, est hérité de la philosophie d'Unix que l'on peut ensuite retrouver dans les logiciels libres en général. Ce qui est d'autant plus intéressant, c'est que les codes de ces logiciels peuvent être consultés et modifiés, ainsi, il est facile de les faire communiquer les uns avec les autres. Il s'agit donc ici de l'antithèse d'un logiciel propriétaire comme Cubase par exemple qui doit permettre de tout faire.

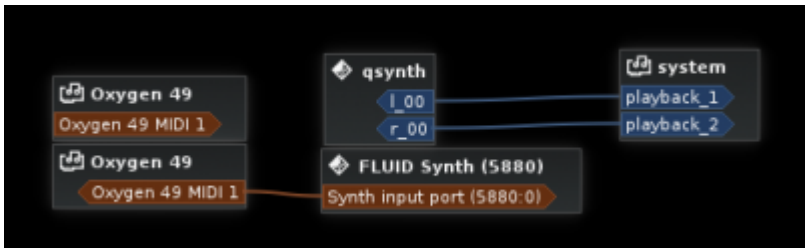
Ainsi, sous Gnu/Linux, Jack [(5)] permet de faire communiquer les applications entre elles en audio comme en MIDI. Diverses interfaces graphiques permettent de visualiser ces connexions et du coup d'avoir un aperçu très précis de ce qui se passe concernant les différents flux. Parmi ces GUI nous avons Catia [(6)] qui s'apparente à un patch Max/MSP ou PureData. On peut directement lier les applications entre elles et bien faire la différence entre ce qui concerne l'audio et le MIDI (les connections bleu symbolisent un flux audio et celles en orange ou rouge les messages MIDI Alsa ou Jack).

Si je veux, par exemple, parler de canaux, ou de messages MIDI, il est très difficile de le faire avec de gros logiciels, par contre si je le fais avec Jack, Carla et Qsynth [(7)] cela va devenir plus clair.

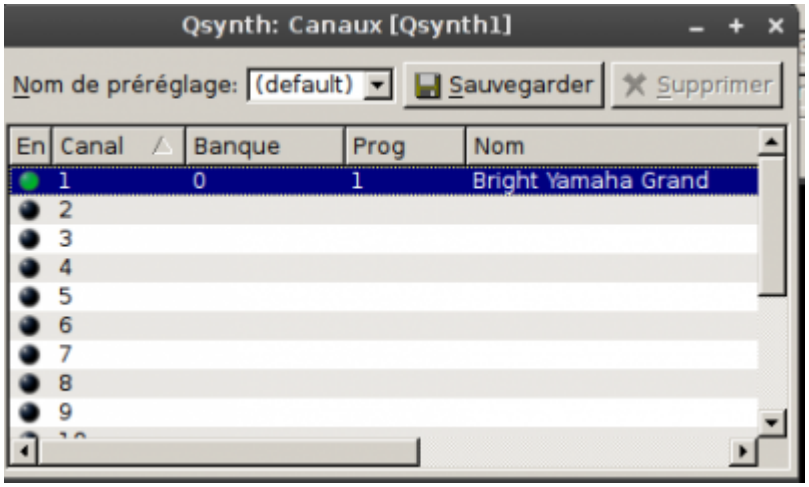
Voici un patch où l'on visualise ces différents éléments :



On voit ici de gauche à droite, notre clavier avec nos entrées et sorties MIDI, le logiciel Qsynth (interface graphique de FluidSynth [8] lecteur de *soundfonts*) avec ses sorties audio stéréo, l'entrée MIDI de FluidSynth et l'entrée dans le système audio de l'ordinateur. Il suffit de lier tout cela pour avoir du son :



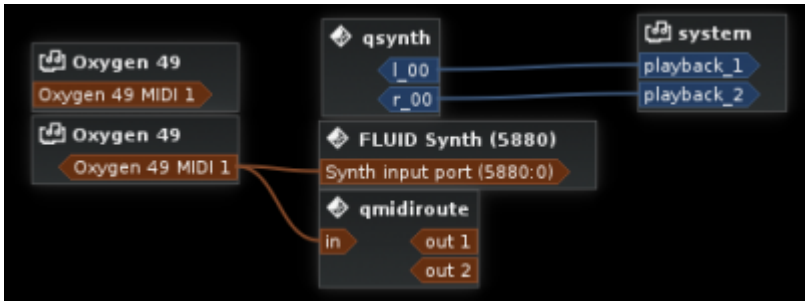
D'autre part on peut facilement voir ce qui se passe, lorsque l'on joue les sons avec le clavier MIDI par l'intermédiaire de Qsynth :



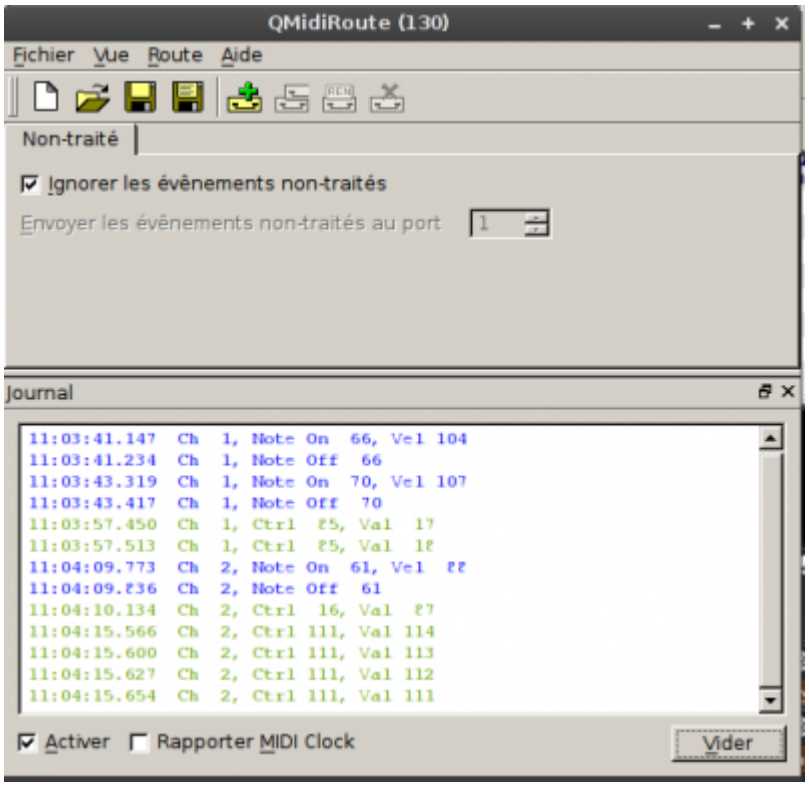
Si l'on joue des notes sur le clavier MIDI, on voit que seule la LED du canal 1 s'allume. Il est donc ici très facile d'appréhender la notion de canal MIDI en changeant le numéro de celui-ci sur le clavier.

Sur le patch, on voit très clairement de façon générale que l'on ne peut pas connecter du MIDI dans de l'audio, et donc de dire que ce ne sont pas du tout les mêmes informations qui transitent.

Nous pouvons aller encore plus loin avec une autre application qui se nomme Qmidiroute [9] et qui permet de visualiser de façon claire tout ce qui sort de notre clavier MIDI. Nous faisons nos connections de la même manière que précédemment :



Ensuite tout ce que nous allons faire sur notre clavier va être écrit :



On voit ici que tout a été pris en compte, c'est-à-dire le numéro du canal, le type d'évènement, le numéro de celui-ci ainsi que sa valeur. Les choses apparaissent ici avec une acuité toute particulière et cela permet de les commenter en cours.

Conclusions

À travers ce texte, il me semble donc clair que plus l'application est simple, plus il est facile de montrer de manière pédagogique les différents éléments d'un langage ou de processus inhérents à l'audio et au MIDI. Certes, les grosses applications propriétaires peuvent avoir leurs places dans la pédagogie si l'on décide d'orienter le cours sur la créativité par exemple, mais les logiciels libres sont irremplaçables pour démontrer ce qui se passe de manière claire dans le système.

Il me paraît ensuite évident de souligner que le plus important n'est pas de savoir utiliser un logiciel, mais de comprendre comment l'audio et le MIDI fonctionnent pour pouvoir utiliser ses compétences sur tout type d'application audio. Ainsi utiliser le plus de logiciels différents possible permet une compréhension profonde de ce que l'on est en train de faire.

---

1. *Graphical User Interface*.
2. http://lilypond.org/
3. http://www.tunefl.com/
4. http://lilypond.org/doc/v2.18/Documentation/learning/index.fr.html
5. http://jackaudio.org/
6. http://kxstudio.sourceforge.net/Applications:Catia

7. http://qsynth.sourceforge.net/qsynth-index.html

8. http://sourceforge.net/apps/trac/fluidsynth/

9. http://apps.linuxaudio.org/apps/all/qmidiroute

Pour citer ce document:
Lionel Rascle, « La pédagogie face à l'informatique musicale », *RFIM* [En ligne], Numéros, n° 4 - automne 2014, Mis à jour le 24/09/2014
URL: http://revues.mshparisnord.org/rfim/index.php?id=273