

## Feuille de travaux pratiques

### Arpèges

#### Exercice 1 (Arpèges)

Dans cet exercice, on s'intéresse à la génération d'arpèges. Un arpège<sup>1</sup> est un accord, c'est-à-dire un groupe de notes sonnant harmonieusement quand elles sont jouées ensemble, mais dont les notes sont jouées successivement. En général, les notes sont jouées dans un ordre croissant, décroissant, ou d'abord croissant puis décroissant. On considèrera qu'il y a un intervalle de temps constant entre chaque note jouée.

On veut ici générer des arpèges en calculant les hauteurs MIDI des notes jouées successivement. Cela dépend de l'accord que l'on veut arpégier : dans la suite, cet accord sera donné par trois notes, encodées de la façon suivante :

- une note basse (par exemple do 3, ou 60 en hauteur MIDI),
- l'intervalle entre la note basse et la deuxième note de l'accord (par exemple une tierce majeure, ou 4 en MIDI)
- l'intervalle entre la note basse et la troisième note de l'accord (par exemple une quinte juste, ou 7 en MIDI).

Cet encodage est assez proche d'un chiffrage d'accord, notion couramment utilisée aussi bien en musique occidentale classique qu'en musiques actuelles, et qui permet de déconnecter l'accord lui-même (décrit par les intervalles) de la hauteur à laquelle il est joué (donné par la note basse). Par exemple, voici quelques exemples d'accords donnés par un même chiffrage, celui d'accord parfait majeur :

Noms des notes	hauteurs MIDI	note basse	intervalles
do3 - mi3 - sol3	60 64 67	60	4 7
do4 - mi4 - sol4	72 76 79	72	4 7
sol3 - si3 - re4	67 71 74	67	4 7

A partir d'une note basse, que l'on notera `basse` dans la suite, et de deux intervalles `itv1` et `itv2`, on jouera naturellement un arpège de quatre notes déterminées ainsi : `basse`, `basse + itv1`, `basse + itv2`, `basse + 12` (on finit en général par la note basse, jouée un octave plus haut). On peut répéter ce principe sur plusieurs octaves. On veut en outre enchaîner un arpège montant avec le même arpège descendant. Voici par exemple la suite de notes correspondant à `basse = 48`, `itv1 = 4`, `itv2 = 7` sur trois octaves :

48 – 52 – 55 – 60 – 64 – 67 – 72 – 76 – 79 – 84 – 84 – 79 – 76 – 72 – 67 – 64 – 60 – 55 – 52 – 48

1. Pour écouter, voir par exemple <https://www.youtube.com/watch?v=VtEOxEckfEc> ou <http://www.six-cordes.com/cours-guitare-debutant/premiers-pas/arpages-accords-la-maj/>.

Cet exercice peut être fait, selon votre aisance en programmation fonctionnelle, de deux façons : soit directement (excellent exercice de modélisation et récursion), en écrivant directement une fonction `arpegie` qui calcule les valeurs successives des notes, et les imprime à la volée, soit avec les indications suivantes :

1. Ecrire une fonction `gere_note` qui prend en entrée une note au format MIDI et l'imprime. NB : en modifiant cette fonction, on pourra selon les besoins afficher la note au format MIDI, l'afficher au format note de musique (cf Exercice 2 feuille de TD 2), ou encore la jouer).
2. déterminer le type de la fonction `arpegie`.
3. déterminer le(s) cas de base de la fonction `arpegie` : cette fonction a deux régimes, arpège montant et arpège descendant. Comment gérer cela ? Quels sont les cas d'arrêt de chacun des régimes ?
4. à un instant donné, on suppose que la fonction `arpegie` joue dans l'arpège montant une note `note_courante`. Déterminer le calcul qui permet de calculer la note suivante, en distinguant bien les cas.
5. Ecrire une fonction `montant` qui calcule les notes de l'arpège montant (cette fonction pourra ensuite être locale à `arpegie`).
6. Déterminer comment utiliser la même récursion pour les arpèges descendantes.
7. Ecrire la fonction `arpegie`.