

## Devoir de programmation

### I. Harmonisation automatique

L'objet du projet est de réaliser l'harmonisation d'un chant donné en utilisant les règles de l'harmonie classique. On se donne une mélodie (soprano), il s'agit de construire les trois lignes mélodiques (alto, ténor, basse) en respectant un jeu simplifié de règles décrite dans le manuel *Vingt leçons d'harmonie* de Jean-Louis Foucart. Nous chercherons dans ce projet à produire de belles harmonisations. On s'efforcera, par exemple, à minimiser les intervalles dans les parties intermédiaires (alto et ténor) en privilégiant les notes tenues. En revanche, on pourra chercher à avoir une ligne de basse de grand amplitude.

Pour simplifier le travail, on travaillera sur une mélodie sans modulation (pas de bémol et de dièse) en Do majeur et les durées des notes seront des multiples d'un temps. La partition Fig.1 donne un exemple d'harmonisation du début du chant «Au clair de la lune» : do do do ré mi re do mi re re do.



Fig1. Harmonisation de «Au clair de la lune»

### II. Notes, chants, accords et premiers éléments d'harmonie

Plusieurs systèmes sont couramment utilisés pour noter les notes (voir Notation des hauteurs de notes). Vous serez amené à les utiliser pour produire à votre tour des partitions et des fichiers de musique au format midi. Cependant, dans un premier temps, je vous propose un format simple adapté à l'application des règles d'harmonie qui ne nécessite pas de connaissance préalable de la musique. Nous considérons que les notes sont représentées par des entiers compris entre 0 et 27. La table suivante associe à chaque entier, le nom et le numéro d'octave de la note.

octave	do	ré	mi	fa	sol	la	si
1	0	1	2	3	4	5	6
2	7	8	9	10	11	12	13
3	14	15	16	17	18	19	20
4	21	22	23	24	25	26	27

Dans notre système, au clair de la lune s'écrit de deux manières :

1. do4 do4 do4 re4 mi4 re4 do4 mi4 re4 re4 do4
2. 21 21 21 22 23 22 21 23 22 22 21

Nous considérons qu'un chant est une suite de notes enrichis d'une durée au format «note\_durée»

1. do4:1 do4:1 do4:1 ré4:1 mi4:2 re4:2 do4:1 mi4:1 re4:1 re4:1 do4:3 -:1

Pour faciliter la compréhension d'un chant, nous préférons la notation (nom,octave). Dans notre chant, nous avons introduit une nouvelle note «-» représentant un silence.

Nous appelons accord une combinaison de trois notes ordonnées sans précision de l'octave. La première note est appelé tonique et les deux suivantes tierce et quinte. Chaque accord sera noté par un chiffre romain. Le tableau suivant donne la liste des accords utilisés par notre technique d'harmonisation :

Accord	Tonique	Tierce	quinte
I	do	mi	sol
II	ré	fa	la
III	mi	sol	si
IV et IVb	fa	la	do
V	sol	si	ré
VI	la	do	mi
VII	si	ré	fa

Nous avons introduit deux type d'accord IV (IV et IVb) pour simplifier les règles d'enchaînement d'accords.

Dans notre système, harmoniser un chant revient à calculer à chaque temps une note pour l'alto, le ténor et la basse enrichi d'un accord. Lorsqu'à un temps, le chant ne joue pas de note, la note considérée sera la dernière note jouée. Le tableau suivant donne la transcription de l'harmonisation de «Au clair de la lune». Nous noterons qu'à chaque temps, tous les notes de l'accord sont joués par le ténor, l'alto et le soprano, et que la basse reprend la tonique de l'accord.

Soprano	do <sub>4</sub> 21	do <sub>4</sub> 21	do <sub>4</sub> 21	ré <sub>4</sub> 22	mi <sub>4</sub> 23		ré <sub>4</sub> 22		do <sub>4</sub> 21	mi <sub>4</sub> 23	ré <sub>4</sub> 22	ré <sub>4</sub> 22	do <sub>4</sub> 21			-
Alto	sol <sub>3</sub> 18	la <sub>3</sub> 19	la <sub>3</sub> 19	si <sub>3</sub> 20	do <sub>4</sub> 21	si <sub>3</sub> 20	la <sub>3</sub> 19	si <sub>3</sub> 20	la <sub>3</sub> 19	do <sub>4</sub> 22	si <sub>3</sub> 20	si <sub>3</sub> 20	la <sub>3</sub> 19	sol <sub>3</sub> 18	la <sub>3</sub> 19	sol <sub>3</sub> 18
Ténor	mi <sub>3</sub> 16	fa <sub>3</sub> 17	mi <sub>3</sub> 16	sol <sub>3</sub> 18	sol <sub>3</sub> 18	sol <sub>3</sub> 18	fa <sub>3</sub> 17	sol <sub>3</sub> 18	fa <sub>3</sub> 17	sol <sub>3</sub> 18	sol <sub>3</sub> 18	sol <sub>3</sub> 18	fa <sub>3</sub> 17	mi <sub>3</sub> 16	fa <sub>3</sub> 17	mi <sub>3</sub> 16
Basse	do <sub>2</sub> 7	fa <sub>2</sub> 10	la <sub>2</sub> 12	sol <sub>2</sub> 13	do <sub>2</sub> 7	mi <sub>2</sub> 9	ré <sub>2</sub> 8	sol <sub>2</sub> 13	fa <sub>2</sub> 10	do <sub>2</sub> 7	sol <sub>2</sub> 13	sol <sub>2</sub> 13	fa <sub>2</sub> 10	do <sub>2</sub> 7	fa <sub>2</sub> 10	do <sub>2</sub> 7
Accord	I	IV	VI	V	I	III	II	V	IVb	I	V	V	IVb	I	IV	I

### III. Un jeu simple de règles d'harmonie

Dans cette partie, nous nous contenterons d'énoncer les règles que doivent respecter une harmonisation d'un chant. Nous distinguerons deux types de règles : local et d'enchaînement. Les règles locales s'appliquent aux notes des quatre chants à un temps donnée et les règles d'enchaînement concernent les contraintes des chants à un instant et l'instant suivant.

**Règles 1 (local) :** Les notes des voix doivent être comprises entre les valeurs données par le tableau suivant :

	minimum	maximum
<b>Soprano</b>	do3 (14)	la4 (26)
<b>Alto</b>	sol2 (11)	ré4 (22)
<b>Ténor</b>	do2 (7)	la3 (19)
<b>Basse</b>	fa1 (3)	ré3 (15)

**Règles 2 (local) :** A chaque instant, la note de basse est plus petite que celle du ténor, celle du ténor plus petit que celle de l'alto et celle de l'alto plus petit que celle du soprano.

**Règles 3 (local) :** A chaque instant, la note de basse est la tonique de l'accord et les notes des 3 autres chants doivent recomposer exactement l'accord (sans nécessairement respecter l'ordre)

**Règles 4 (local) :** La suite d'accord ne peut pas commencer ou finir par l'accord IVb.

**Règles 5 (enchaînement) :** A chaque instant les accords doivent respecter les règles d'enchaînement donnée par la table suivante :

Cet accord	s'enchaîne	vers
I	→	Tous sauf IVb
II	→	V ou VII
III	→	Tous sauf I et IVb
iV	→	Tous sauf IVb
iVb	→	I
V	→	I, III, VI ou IVb
VI	→	II, III, IV, V
VII	→	I, III

**Règles 6 (enchaînement) :** A un instant donnée et pour un chant donnée, la différence entre les deux notes ne peut pas excéder 6. De plus, si la note appartient à l'accord suivant, elle ne doit pas

changer. Quand la différence entre les deux notes excèdent 2 (strictement), elles doivent être de même nature (tonique, tierce ou quinte) dans leurs accords respectifs.

#### IV. Calcul de tous les harmonisations d'un chant

**Etape 1 :** Dans un premier temps, on cherchera à coder un jeu de notes <soprano,alto,ténor,base,accord> de la manière la plus économe que possible (i.e un int) pour se prémunir partiellement des problèmes de mémoire.

**Etape 2 :** Pour une note de soprano donnée, être capable de calculer sous la forme d'un tableau, tous les jeux de notes respectant les règles d'harmonisation local.

**Etape 3 :** Pour un chant donné, calculer un tableau associant à chaque note du chant, la liste des jeux calculée à l'étape 2.

**Etape 4 :** Etre capable de tester le respect des règles d'enchaînement entre deux jeux de notes.

**Etape 5 :** Pour un chant donné, calculer pour chaque note et chaque jeu de notes, la liste des jeux à l'instant suivant respectant les règles d'enchaînement. Dans ces listes, les jeux suivants seront identifiés par leur indice dans leur tableau de jeux.

A cette étape, vous avez produit 2 tableaux à deux dimensions :

a. `int[][] jeu avec jeu[i][j]` donne un jeu de notes possible pour l'instant `i`

b. `Liste[][] suivant avec suivant[i][j]` donne la liste des suivants possibles de `jeu[i][j]` sous la forme d'un indice dans le tableau `jeu[i+1]`

**Etape 6 :** Eliminer dans la liste des suivants, les jeux ne produisant aucune harmonisation. Ce calcul peut être réalisé à partir de l'instant juste avant la fin du morceau et en remontant dans le morceau : il suffit de retirer des listes les jeux n'ayant pas de suivant.

**Etape 7 :** Produire une liste des jeux du début du morceaux produisant au moins une harmonisation du chant.

Pour produire un morceau, il suffit de partir d'un jeu calculé à l'étape 7 et suivre un chemin dans les jeux définis par la table des suivants.

#### V. Calcul du nombre d'harmonisations et génération de belles harmonies

Le nombre d'harmonisations à partir d'un instant `i` et d'un `jeu[i][j]` est en général la somme du nombre des harmonisations à partir de l'instant `i+1` des jeux suivants de `jeu[i][j]`. Cette remarque induit un calcul simple et efficace du calcul du nombre d'harmonisations d'un chant. La technique consiste à construire un tableau à deux dimensions `nombre[][]` associant à chaque jeu sont d'harmonisation. Ce tableau pourra être rempli en commençant par la fin du chant. En particulier, les valeurs des jeux pour la fin du chant seront initialisé à 1 et le nombre d'harmonisations du chant sera égal à la somme des `nombre[0][j]`.

Ce même principe peut être utilisé pour la génération de belles harmonisations. Une fonction de beauté locale sera vue comme une fonction associant à un jeu, un entier. Plus la valeur de la fonction est grande, plus le jeu est beau. La beauté d'un arrangement sera donné comme la somme de la beauté des jeux qui la composent. Pour générer de belles harmonies, il suffit de calculer pour chaque `jeu[i][j]` la valeur de la plus belle harmonisation d'une manière similaire à celle calculer le nombre d'harmonisation, et de suivre dans la génération les `jeu[i][j]` ayant les plus grandes valeurs.

La généralisation de la génération de belles harmonies à partir d'une fonction de beauté d'enchaînement est laissée à votre charge. Nous considérons qu'une fonction de beauté d'enchaînement associe à un couple de jeux un entier. La beauté d'un arrangement est la somme de la beauté des enchaînements qui la composent.

Dans votre projet, vous serez amené à étudier 3 critères élémentaires de beautés :

1. Une fonction privilégiant un faible écart entre l'alto et le ténor
2. Une fonction privilégiant la tenue de notes de l'alto et du ténor
3. Une fonction privilégiant une grande amplitude de la basse

Vous serez aussi amené à proposer un critère combinant les trois critères précédents.

## VI. Le sujet

Réaliser un programme d'harmonisation en Java permettant de

- lire un chant à partir de fichiers dans un format simple :
  - La première ligne contient le titre du morceau,
  - La deuxième ligne, la suite des notes enrichies d'une durée (i.e do4:1 do4:1 do4:1 re4:1 mi4:2 re4:2 do4:1 mi4:1 re4:1 re4:1 do4:3 -:1)
- Calculer tous les harmonisations d'un chant et en calculer leurs nombres.
- Produire une harmonisation d'un chant au format LiLyPond (type partition) et au format midi (voir table de correspondance note-midi)
- Produire de belles harmonisations en fonction des 4 critères décrits à la section précédente.
- Traiter un ensemble de chants et produire les résultats des calculs sous la forme d'une page html.

	do	ré	mi	fa	sol	la	si
1	36	38	40	41	43	45	47
2	48	50	52	53	55	57	59
3	60	62	64	65	67	69	71
4	72	74	76	77	79	81	83

Table donnant à chaque note son format midi

## VII. Que doit faire votre programme ?

Le programme java sera composé d'un unique archive jar **Harmonie.jar**. Ce programme devra s'exécuter dans un terminal avec les options suivantes :

1. `java -jar Harmonie.jar -name` affiche vos noms et prénoms
2. `java -jar Harmonie.jar -h` rappelle la liste des options du programme
3. `java -jar Harmonie.jar -midi fichier1.chant fichier2.mid` donne une harmonisation du chant contenu dans `fichier1.chant` sous la forme d'un fichier midi `fichier2.mid`.

4. `java -jar Harmonie.jar -ly fichier1.chant fichier2.mid` donne une harmonisation du chant contenu dans `fichier1.chant` sous la forme d'un fichier LiLyPond `fichier2.ly`.
5. `java -jar Harmonie.jar -nombre fichier1.chant` écrit le nombre d'harmonisations du chant contenu dans `fichier1.chant`.
6. `java -jar Harmonie.jar -beaute k -midi fichier1.chant fichier2.mid` donne une plus belle harmonisation du chant contenu dans `fichier1.chant` suivant le critère de beauté numéro `k` ( $k=1..4$ ) sous la forme d'un fichier midi `fichier2.mid`.
7. `java -jar Harmonie.jar -beaute k -ly fichier1.chant fichier2.mid` donne une plus belle harmonisation du chant contenu dans `fichier1.chant` suivant le critère de beauté numéro `k` ( $k=1..4$ ) sous la forme d'un fichier LiLyPond `fichier2.ly`.
8. `java -jar Harmonie.jar -w dossier1 dossier2` donne le résultat des calculs demandés précédents (3 à 7) appliqués à tous les fichiers de chant du dossier `dossier1` sous la forme d'un dossier `dossier2` contenant les fichiers midi et LiLyPond calculés et une page html produisant un tableau qui, pour chaque chant, donne le nom, le nombre d'harmonisations et les liens sur les fichiers calculés.

## VIII. Quoi et quand rendre ?

1. Quand ? **le mercredi 30 avril 2014 à 12h00**
2. Envoyer par email à votre chargé de TD, votre rapport au format pdf, votre fichier `Harmonie.jar` et un dossier compressé contenant vos fichiers de chants utilisés pour votre expérimentation.
3. Donner à votre chargé de TD en version papier, votre rapport de projet. Ce rapport sera en trois parties : (1) Travail réalisé, (2) Architecture et (3) Expérimentation.

## IX. Quelles références disponibles sur la toile et à la bibliothèque des sciences

4. <http://java.sun.com/javase/6/docs/api/index.html> API java
5. <http://java.sun.com/docs/codeconv/index.html> donne les conventions d'écriture d'un programme java
6. <http://java.sun.com/j2se/javadoc/writingdoccomments/index.html> donne un ensemble d'informations suffisantes pour l'écriture de votre rapport de projet sous la forme de commentaires utilisables par l'outil javadoc.
7. [La programmation en pratique](#), Brian W. Kernigan et Rob Pike. Editeur Vuibert informatique. Lire le premier chapitre sur la manière de commenter un programme et choisir des noms de variables
8. [Au coeur de Java 2 \(Volume 1\), Notions Fondamentales](#), Cay S. Horstmann et Gary Cornell. Editeur CampusPress. Livre recommandé.
9. Notation des hauteurs de notes, [www.crlm.paris4.sorbonne.fr/organologie/annexe\\_1.pdf](http://www.crlm.paris4.sorbonne.fr/organologie/annexe_1.pdf)
10. L'outil de notation musicale Lilypond, <http://lilypond.org/>
11. Documents de l'outil LiLyPond, <http://lilypond.org/doc/v2.12/Documentation/>
12. Vingt leçons d'harmonisation, Jean-Louis Foucart, <http://www.musiccomposer.fr/static/extraits.pdf>