
TP 1

COMPOSITION MUSICALE

Introduction

Pour le premier devoir, vous allez construire un logiciel qui compose une mélodie. Ce logiciel va demander des informations de base sur la mélodie à composer : accord de base, durée et type de progression. La composition sera semi-aléatoire. Elle sera basée sur un texte choisi par l'utilisateur. Les sections suivantes vont décrire les entrées, le fonctionnement et la sortie du logiciel. Finalement, les modalités de remise et d'évaluation sont présentées.

Description des entrées

Les entrées du logiciel seront données par l'utilisateur sous la forme d'une interaction au clavier. Votre logiciel devra poser cinq (5) questions à l'utilisateur. Les questions doivent être posées dans l'ordre présenté ici.

1. Demandez le nom d'un fichier texte (.txt) à l'utilisateur. Quand l'utilisateur écrit le nom du fichier, c'est sa responsabilité d'écrire l'extension du fichier. Ce fichier sera utilisé comme une suite de caractères. La lettre *S* sera utilisée pour représenter cette séquence dans le reste de cette description.
2. Demandez le type de progression qui sera utilisé par le logiciel. L'utilisateur doit choisir entre Années 50, Blues, Circulaire, Mineur ou Populaire. L'utilisateur ne fait qu'entrer la première lettre du mot, en majuscule ou en minuscule : A, B, C, M ou P. Une progression représente une suite d'accord. Dans ce texte, chaque accord est représenté par un nombre romain, soit en majuscule, soit en minuscule. Selon la progression choisie, nous obtiendrons la suite suivante :

<i>Progression</i>	<i>Suite d'accords</i>
<i>Année 50</i>	I, vi, ii, V
<i>Blues</i>	I, V, IV, IV, I, V, I, V
<i>Circulaire</i>	I, IV, V, I
<i>Mineur</i>	I, VII, VI, V
<i>Populaire</i>	I, IV, V, V

3. Demandez la tonalité. La tonalité est une lettre entre A et G. Elle peut être majuscule ou minuscule. Cette tonalité devra être transformée en un nombre *d* selon la table de conversion suivante :

<i>Tonalité</i>	<i>Nombre d</i>
<i>C</i>	0
<i>D</i>	2
<i>E</i>	4
<i>F</i>	5
<i>G</i>	7
<i>A</i>	9
<i>B</i>	11

4. Demandez le numéro t d'octave. Cette valeur doit être entre 0 et 7.
5. Finalement, demander le nombre de mesures n que contiendra la mélodie. Cette valeur doit être plus grande que 1. Une mesure est une unité temporelle relative en musique. En général, chaque mesure dure le même temps. Par exemple, nous pouvons décider qu'une mesure dure 2 secondes, alors si notre mélodie contient 130 mesures, elle durera $130 \times 2 = 260$ secondes, ou 4 minutes et 20 secondes.

Construction de la mélodie

La construction de la mélodie consiste à construire une séquence de base qui sera ensuite transformée à plusieurs reprises pour obtenir la séquence finale (la mélodie). Les transformations d'une séquence à l'autre seront dirigées par la séquence S de caractères contenus dans le fichier en entrées. Voici les grandes étapes :

1. Construire le rythme (beat) de base.
2. Construire une séquence de temps pour la mélodie.
3. Construire une table de progression.
4. Transformer la séquence de temps en mélodie.
5. Transformer la mélodie en format de sortie pour le logiciel.

Utilisation de la séquence de caractères

Les étapes 1, 2 et 4 vont utiliser la séquence de caractère S afin de simuler des choix aléatoires. Les caractères de la séquence sont lus un à la suite de l'autre. Lorsque l'algorithme demande un caractère de la séquence, il faut trouver le prochain caractère répondant aux caractéristiques demandées. Si un caractère ne répond pas aux caractéristiques demandées, alors il est passé. Nous ne revenons jamais en arrière dans la séquence. Par exemple, si nous cherchons un caractère entre 'a' et 'g' (caractéristique : le caractère est entre 'a' et 'g' inclusivement) dans cette séquence : « transformations ». Alors, les caractères 't' et 'r' seront passés et le caractère 'a' sera retourné. Remarquez que la case des caractères lus n'est pas importante, un 'a' est équivalent à un 'A'.

Algorithmes

Construire le rythme de base

Un rythme indique la durée de chaque note dans la mélodie. Généralement, le rythme se répète tout au long de la musique, avec quelques variations. Votre logiciel va commencer par construire un rythme de base. Pour nos besoins, nous allons diviser une mesure en 16 mini-intervalles temporels égaux (numéroté de 1 à 16). Ces 16 mini-intervalles sont regroupés en 4 paquets de 4 mini-intervalles chacun

[1,2,3,4|5,6,7,8|9,10,11,12|13,14,15,16]. Pour varier ce rythme, nous allons extraire deux caractères de notre séquence S de lettre. La caractéristique de ces caractères est qu'ils doivent être dans l'ensemble suivant : { C, D, F, G, H, I, L, M, N, O, P, R, S, U, W, Y }. Ces deux caractères vont indiquer deux endroits où le rythme va être divisé en deux. Pour cela, nous assignons une lettre à chaque mini-intervalle.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
W	M	U	C	S	N	I	O	L	D	R	H	P	Y	G	F

La première lettre va indiquer le paquet qui doit être divisé en deux paquets égaux. Par exemple, si la lettre 'D' est trouvée, alors le troisième paquet est divisé en deux, nous donnant les paquets : [1,2,3,4|5,6,7,8|9,10|11,12|13,14,15,16]. La deuxième lettre est utilisée de la même façon. Elle va indiquer lequel des cinq paquets sera divisé en deux paquets égaux. Par exemple, si la deuxième lettre est un 'H', alors le quatrième paquet sera divisé en deux, nous donnant les paquets : [1,2,3,4|5,6,7,8|9,10|11|12|13,14,15,16]. Les temps de notre rythme sont mesurés en comptant le nombre de mini-intervalles que nous avons dans nos paquets. Dans notre exemple, nous avons des temps de 4-4-2-1-1-4. Remarquez que cette technique va toujours résulter avec 6 paquets (temps).

Construire la séquence des temps pour la mélodie

Le rythme que nous avons construit indique les temps pour diviser une mesure. Or, notre mélodie sera composée de n mesures (tel qu'indiqué par l'utilisateur). La première mesure va utiliser le rythme de base que nous avons construit. Les autres mesures vont utiliser soit le rythme de base, ou une variation sur le rythme de base. Pour chaque mesure suivant la première, l'algorithme suivant est utilisé pour trouver le rythme de cette mesure.

1. Extraire une lettre de la séquence S ayant comme caractéristique qu'elle est entre 'A' et 'Z'.
2. Si la lettre extraite n'est pas dans l'ensemble {J,X,Q,Z}, alors la mesure utilise le rythme de base
3. Si la lettre extraite est dans l'ensemble {J,X,Q,Z}, alors une modification temporaire est apportée au rythme de base. Cette modification est effectuée comme lors de la construction du rythme de base :
 - a. Une lettre est extraite de la séquence S , elle doit appartenir à l'ensemble { C, D, F, G, H, I, L, M, N, O, P, R, S, U, W, Y }.
 - b. Cette lettre est utilisée pour séparer un paquet en deux paquets égaux. Si le paquet à séparer ne contient qu'un mini-intervalle, alors choisir une autre lettre.
 - c. Le rythme modifié est utilisé pour cette mesure. Remarquez que ce rythme contiendra 7 temps.
 - d. La modification est supprimée, nous revenons au rythme de base pour la prochaine itération.

Ces étapes vont construire une séquence de temps. Cela devrait donner une suite de valeur. Les seules valeurs que cette suite va contenir sont {1,2,4}. Il est important que cette séquence indique, d'une façon ou d'une autre, la séparation entre les mesures.

Construire une table de progression

Avant de générer la mélodie, nous devons construire une table de progression. Une progression est un accord. Un accord est un ensemble de notes. Ces accords dépendent de la tonalité de base (d et t) de la mélodie (choisi par l'utilisateur). Nous allons représenter les notes de notre musique par des valeurs de 0 à 127. La tonalité de base b est une note dont la valeur est calculée par l'équation suivante : $b = 12(t + 1) + d$. Par exemple, si l'utilisateur a demandé une tonalité de 'F' ($d = 5$) de l'octave 4 ($t = 4$), alors la note de base sera $b = 65$.

L'utilisateur à choisi une progression, soit Années 50, Blues, Circulaire, Mineur ou Populaire. Cette progression donne une suite d'accord, représenté par des nombres romains, en majuscule ou en minuscule. Chacun de ces accords est représenté par trois notes. Ces notes sont calculées par rapport à la note de base. La première note de l'accord est décidée par le numéro en Romain qui la représente :

Accord	Décalage de la première note
<i>I</i>	+0
<i>ii</i>	+2
<i>IV</i>	+5
<i>V</i>	+7
<i>VI/vi</i>	+9
<i>VII</i>	+11

La première note de l'accord est construite en ajoutant ce décalage à la note de base. Les deuxième et troisième notes de l'accord sont décidées par la case (minuscule ou majuscule) du nombre romain. Note : les minuscules représentent les accords dits 'mineurs' et les majuscules représentent les accords dits 'majeurs'.

Accord	Deuxième note	Troisième note
<i>Minuscule (ii, vi)</i>	+3	+7
<i>Majuscule (I, IV, V, VI, VII)</i>	+4	+7

Ces décalages sont ajoutés à la première note de l'accord pour construire les deux autres notes. Pour continuer notre exemple, si l'utilisateur a choisi une progression Année 50, avec la tonalité de base de $b = 65$ que nous avons calculé préalablement. Nous obtiendrons la table de progression suivante :

Progression	Première note	Deuxième note	Troisième note
<i>I</i>	$65 + 0 = 65$	$65 + 4 = 69$	$65 + 7 = 72$
<i>vi</i>	$65 + 9 = 74$	$74 + 3 = 77$	$74 + 7 = 81$
<i>ii</i>	$65 + 2 = 67$	$67 + 3 = 70$	$67 + 7 = 74$
<i>V</i>	$65 + 7 = 72$	$72 + 4 = 76$	$72 + 7 = 79$

Transformer la séquence de temps en mélodie

Ensuite, la table de progression est utilisée pour transformer la séquence de temps en mélodie. Chaque note de la mélodie est représentée par une valeur de tonalité (entre 0 et 127) et une durée (entre 1 et 16). Pour construire ces valeurs, nous allons utiliser la table de progression qui nous donnera les tonalités et la séquence des temps, qui nous donnera les durées. La séquence des temps contient aussi le changement des mesures. Ce changement est important, à chaque changement de mesure, nous changeons d'accord dans la table de progression. La première mesure utilise le premier accord de la progression, la deuxième mesure utilise le deuxième accord de la progression, et ainsi de suite. En général, il y aura plus de mesure qu'il y a d'accord

dans la table de progression. Lorsqu'il n'y a plus d'accord dans la table de progression, nous recommençons au début de la table de progression (peut être utilisé le calcul modulaire pour cela!). Votre algorithme va construire des notes jusqu'à ce que tous les temps de la séquence de temps aient été utilisés. Il est à remarquer qu'une note peut utiliser plus de 1 temps de la séquence de temps, elle peut utiliser jusqu'à quatre temps. Voici l'algorithme qui permet de construire une note.

1. Extraire une lettre de la séquence *S*, elle a comme caractéristique qu'elle appartient à l'ensemble {A, I, O}. La lettre 'A' indique que la prochaine note est la première note de l'accord utilisé pour la mesure courante. La lettre 'O' indique que la prochaine note est la deuxième note de l'accord utilisé pour la mesure courante et la lettre 'I' indique que la prochaine note est la troisième note de l'accord utilisé pour la mesure courante.

Lettre	Note
A	Première
O	Deuxième
I	Troisième

2. Extraire une lettre de la séquence *S*, elle doit appartenir à l'ensemble {B, C, D, F, G, H, L, M, N, P, R, S, U, V, W, Y}. Cette lettre indique combien de temps la note doit durer. La table suivante indique le nombre de temps que nous devons extraire de la séquence des temps. Remarque : cette table est différente de celle présentée en classe, et la remplace.

Lettre	Nombre de temps
N, S, H, R	1
D, L, C	2
U, M, W	3
F, G, Y	4
P, B	5
V	6

Par exemple, utilisons la séquence des temps suivante, où les '|' indiquent les changements de mesure.

[4-4-2-1-1-4|4-4-2-1-1-4|4-2-2-2-1-1-4|4-4-2-1-1-4|2-2-4-2-1-1-4]

Nous allons utiliser la table de progression de l'exemple précédant. Les prochains caractères de la séquence *S* sont les suivants : « acifanoyadimovodopaw ». Nous sommes à la première mesure, donc le premier accord (I) de la progression est utilisé. Le caractère 'a' est trouvé, donc la prochaine note est le 65. La lettre suivante est un 'c', donc cette note va durer deux temps, les deux premiers temps de la séquence sont 4 et 4. La note 65 aura une durée de 4 + 4 = 8. Voici une trace de cet algorithme.

Mesure 1, accord I		
'a' : note 65	'c' : 2 temps, $4 + 4 = 8$	65, 8
'i' : note 72	'f' : 4 temps, $2 + 1 + 1 + 4 = 8$	72, 8
Mesure 2, accord vi		
'a' : note 74	'n' : 1 temps, 4	74, 4
'o' : note 77	'y' : 4 temps, $4 + 2 + 1 + 1 = 8$	77, 8
'a' : note 74	'd' : 2 temps, $4 + 4 + 2 = 10$	74, 10
Mesure 3, accord ii		
'i' : note 74	'm' : 3 temps, $2 + 2 + 1 = 5$	74, 5
'o' : note 70	'v' : 6 temps, $1 + 4 + 4 + 4 + 2 + 1 = 16$	70, 16
Mesure 4, accord V		
'o' : note 76	'd' : 2 temps, $1 + 4 = 5$	76, 5
Mesure 5, accord I		
'o' : note 69	'p' : 5 temps, $2 + 2 + 4 + 2 + 1 = 11$	69, 11
'a' : note 65	'w' : 3 temps, $1 + 4 + _ = 5$	65, 5

Remarque : la dernière note demandait 3 temps, mais il n'en restait que 2, alors seulement 2 sont pris. Il est aussi à remarquer que lorsqu'une mesure n'a plus de temps, alors les temps de la mesure suivante sont utilisés. Notez que si la séquence de lettre dans le fichier d'entrées ne contient pas assez de lettres pour composer la mélodie, alors vous affichez un message d'erreur et le programme se termine.

Transformer la mélodie en format de sortie pour le logiciel

Cette transformation est très différente (et plus simple) que celle vue en classe. Votre logiciel va écrire les résultats dans un fichier nommé « melodie.txt ». Les résultats sont affichés à raison d'une note par ligne. Une note est simplement la tonalité de la note, une espace, la durée de la note.

Directive

1. Le tp est à faire seul ou en équipe de deux (maximum).
2. Commentaire :
 - a. Commentez l'entête de chaque méthode. Ces commentaires doivent contenir la description de la fonction et le rôle de ces paramètres.
 - b. Une ligne contient soit un commentaire, soit du code, pas les deux.
 - c. Utilisez des noms d'identificateur significatif.
 - d. Utilisez le français.
3. Code :
 - a. Pas de `goto`, continue.
 - b. Les `break` ne peuvent apparaître que dans les `switch`.
 - c. Un seul `return` par méthode.
4. Indentez votre code. Assurez-vous que l'indentation est faite avec des espaces.

Remise

Remettre le tp par l'entremise de Moodle. Utilisez un fichier compressé de Windows pour construire une archive contenant vos fichiers de code .java. Si vous n'utilisez pas Windows, allez au laboratoire pour construire une archive sur un ordinateur Windows. Le tp est à remettre avant le 9 juin 23:59.

Évaluation+

- Fonctionnalité (8 pts) : Votre tp doit compiler sans erreur (il peut y avoir des warnings). J'utilise Java 8.
- Structure (2 pt) : Il faut avoir **plusieurs** méthodes. Construisez un code bien structuré. Il y a de la place pour quelques classes dans ce programme.
- Lisibilité (4 pts) : commentaire, indentation et noms d'identificateur significatif.