

---

# Rapport de DM

---

TODO

29 décembre 2024

## Table des matières

<b>I</b>	<b>Questions</b>	<b>1</b>
<b>II</b>	<b>Description des fichiers</b>	<b>2</b>
II.1	Description des fichiers sans fonctionnalité supplémentaire . . . . .	2
II.2	Description des fichiers avec fonctionnalité supplémentaire . . . . .	2
<b>III</b>	<b>Description du nouveau format de fichier</b>	<b>2</b>
III.1	Header . . . . .	2
III.2	Piste . . . . .	3
a)	Notes . . . . .	3
b)	Silences . . . . .	4
c)	Commentaires . . . . .	6
III.3	Limitations . . . . .	6
a)	Autres durées de notes . . . . .	6
b)	Accords . . . . .	7
<b>IV</b>	<b>Description de l'ADSR</b>	<b>7</b>
<b>V</b>	<b>Exemple récapitulatif</b>	<b>8</b>
<b>VI</b>	<b>Autres exemples</b>	<b>11</b>
<b>VII</b>	<b>Fioritures</b>	<b>11</b>

## I Questions

**Q1.** La valeur de l'échantillon numéro  $i$  du son sinusoïdal de fréquence  $f$  et d'amplitude  $A$  est :

$$A \sin(2\pi f i \tau_{\text{ech}})$$

**Q3.**

```
1 000000 52 49 46 46 2e 00 00 00 57 41 56 45 66 6d 74 20
2 000010 10 00 00 00 01 00 01 00 22 56 00 00 44 ac 00 00
3 000020 02 00 10 00 64 61 74 61 0a 00 00 00 d2 03 5e 06
4 000030 ff ff a2 f9 ff f7
```

**Q18.** La complexité de la fonction `reduce_mix` est de l'ordre de  $\mathcal{O}(n + n \max(l_1, \dots, l_n))$  car elle contient trois boucles : deux en  $\mathcal{O}(n)$  et une en  $\mathcal{O}(n \max(l_1, \dots, l_n))$ .

## II Description des fichiers

### II.1 Description des fichiers sans fonctionnalité supplémentaire

À part `main.c` qui se situe à la racine, tous les fichiers `.c` et `.h` se situent dans `src/`. Voici la description de la structure du projet :

1. `src/constants.h` : fichier de header contenant toutes les constantes utilisées par plusieurs fichiers. En réalité, ce fichier ne contient que la constante `FREQ_ECH` contenant la valeur de la fréquence d'échantillonnage.
2. `src/melody.c` : permet de lire un fichier contenant une musique.
3. `src/run_tests.c` : la fonction exécutant tous les tests du programme. Cette fonction est appelée par `main.c`. Note au passage : ce programme ne peut tourner que sur des machines POSIX (Unix ou GNU<sup>1</sup>), à cause du fait que `src/run_tests.c` écrit des fichiers dans `/tmp`.
4. `src/sound.c` : fichier lié à la gestion des sons (`sound_t`) : de la création avec les fonctions des différents signaux à la réduction de plusieurs sons en un seul.
5. `src/wav.c` : fichier chargé d'écrire les fichiers WAV : le header et chacun des échantillons d'un son. Contient aussi la fonction `write_int`.

### II.2 Description des fichiers avec fonctionnalité supplémentaire

Par rapport à la version sans fonctionnalité supplémentaire, un seul fichier a été créé : `new_melody.c`, qui contient le code nécessaire pour la lecture du nouveau format (cf. plus loin). Le fichier `melody.c` a été modifié pour détecter quel format est utilisé et donc quelle fonction appeler. Les fichiers `sound.c` et `medoly.c` ont également été modifiés pour permettre l'utilisation d'une enveloppe ADSR.

## III Description du nouveau format de fichier

Ma première fonctionnalité supplémentaire permet l'utilisation d'un nouveau format de fichier plus intuitif, permettant d'écrire le nom et la durée des notes en toutes lettres. **L'ancien format est toujours compatible.**

### III.1 Header

Pour garantir la compatibilité avec l'ancien format et pour permettre de distinguer l'ancien et le nouveau format, **le nouveau format de fichier commence par une ligne contenant un nombre négatif** (n'importe lequel) (ce nombre était le nombre de pistes dans l'ancien format).

La deuxième ligne correspond au **tempo du morceau**, en battement par minute (à la noire). Par exemple, un tempo de  $\text{♩} = 96$  sera transcrit par le header suivant :

```
1 -1
2 tempo 96
```

La ligne suivante contient le nombre de pistes du morceau, et la ligne d'après contient le volume de chaque piste. Par exemple, le fichier `examples/nouveau_format/sonata.txt`, de tempo  $\text{♩} = 130$  et contenant six pistes, commence ainsi :

```
1 -1
2 tempo 130
3 6
4 1 1 1 1 1 1
```

---

1. Indépendamment du noyau (Linux ou Hurd)

## III.2 Piste

Chaque piste commence, comme dans l'ancien format, avec le nombre de notes et l'instrument utilisé. Les instruments sont : **sine**, **square**, **sawtooth**, **triangle**, **sineadsr**, **squareadsr**, **sawtoothadsr** et **triangleadsr** (cf. plus loin pour la description de l'ADSR).

Par exemple, une piste de signal carré comprenant 26 notes commence ainsi :

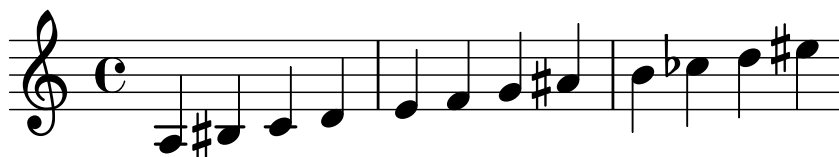
1 26 square

Cependant, contrairement à l'ancien format, les notes ne sont plus au format pitch, durée et volume.

### a) Notes

Une ligne transcrivant une note est composé du **nom de la note**, de **sa durée par rapport au tempo** et de **son volume**.

**Nom de la note** Le nom d'une note est composé de 3 parties (dont deux optionnelles). D'abord, son nom à proprement parler, parmi **do**, **re**, **mi**, **fa**, **sol**, **la**, **si**. Ensuite, il est possible d'indiquer une altération, parmi **b** (pour bémol) et **#** (pour dièse). Enfin, il convient d'indiquer l'octave de la note. L'octave 0 correspond au centre du piano (donc **do0** correspond au do central), l'octave 1 à l'octave au-dessus, l'octave -1 à l'octave en-dessous, etc. *Note : en termes de pitch, ajouter une octave correspond à ajouter 12. De plus, la note de pitch 0 est le la central, i.e. le la de l'octave 0.* Il n'est pas utile de préciser l'octave si l'octave est 0. Par exemple, la portée suivante :



sera transcrite par les notes **la-1**, **si#-1**, **do**, **re**, **mi**, **fa**, **sol**, **la#**, **si**, **dob1**, **re1**, **mi#1**. Une remarque au passage : une note dièse correspond à la note suivante bémol, excepté le mi dièse qui correspond au fa (donc le fa bémol correspond au mi), et de même pour le si dièse qui correspond au do. Ainsi, la portée précédente pouvait aussi, de façon équivalente, être transcrite par les notes **la-1**, **do**, **do**, **re**, **mi**, **fa**, **sol**, **sib**, **si**, **si**, **re1**, **fa1**. En termes de *pitch*, altérer une note correspond à ajouter (dièse) ou soustraire (bémol) 1 au *pitch*.

**Durée de la note** La durée d'une note est écrite en toutes lettres et est dépendante du tempo. Les notes disponibles sont :

1. **ronde** (♩). Durée : 4 temps.
2. **blanche** (♪). Durée : 2 temps.
3. **noire** (♫). Durée : 1 temps.
4. **croche** (♬). Durée :  $\frac{1}{2}$  temps.
5. **doublecroche** (♭). Durée :  $\frac{1}{4}$  temps.
6. **triplecroche** (♮). Durée :  $\frac{1}{8}$  temps.
7. **quadruplecroche** (♯). Durée :  $\frac{1}{16}$  temps.
8. **quintuplecroche** (♭). Durée :  $\frac{1}{32}$  temps.

Les variantes pointées sont également disponibles. Pointer une note lui ajoute la moitié de sa durée ( $\times 1,5$ ). Voici la liste des variantes pointées :

9. **rondepointee** (♩). Durée : 6 temps.
10. **blanchepointee** (♪). Durée : 3 temps.

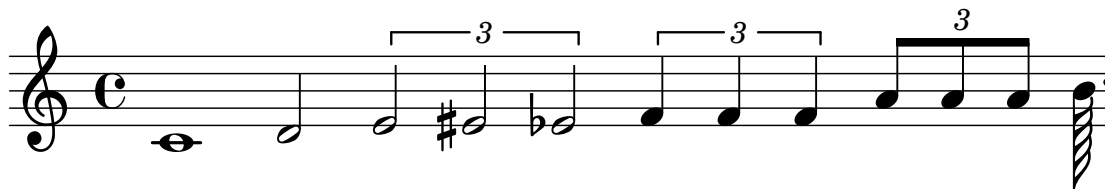
11. **noirepointee** (♩.). Durée :  $\frac{3}{2}$  temps.
12. **crochepointee** (♪.). Durée :  $\frac{3}{4}$  temps.
13. **doublecrochepointee** (♫.). Durée :  $\frac{3}{8}$  temps.
14. **triplecrochepointee** (♬.). Durée :  $\frac{3}{16}$  temps.
15. **quadruplecrochepointee** (♭.). Durée :  $\frac{3}{32}$  temps.
16. **quintuplecrochepointee** (♭.). Durée :  $\frac{3}{64}$  temps.

Enfin, les triolets sont également disponibles. Un triolet d'une certaine note vaut un tiers de la note directement supérieure (par exemple un triolet de croche vaut un tiers de noire). En voici la liste :

17. **trioletderonde**. Durée :  $\frac{8}{3}$  temps.
18. **trioletdeblanche**. Durée :  $\frac{4}{3}$  temps.
19. **trioletdenoire**. Durée :  $\frac{2}{3}$  temps.
20. **trioletdecroche**. Durée :  $\frac{1}{3}$  temps.
21. **triolet**. Alias de **trioletdecroche**.
22. **trioletdedoublecroche**. Durée :  $\frac{1}{6}$  temps.
23. **trioletdetriplecroche**. Durée :  $\frac{1}{12}$  temps.
24. **trioletdequadruplecroche**. Durée :  $\frac{1}{24}$  temps.
25. **trioletdequintuplecroche**. Durée :  $\frac{1}{48}$  temps.

**Volume** Enfin, la note est suivie de son volume, comme dans l'ancien format. Le volume est un flottant entre 0 et 1.

**Exemple** Par exemple, la portée suivante :



sera transcrite par les notes suivantes (avec un volume de 1 à chaque fois) :

```

1 do ronde 1
2 re blanche 1
3 mi trioletdeblanche 1
4 mi# trioletdeblanche 1
5 mib trioletdeblanche 1
6 fa trioletdenoire 1
7 fa trioletdenoire 1
8 fa trioletdenoire 1
9 la triolet 1
10 la triolet 1
11 la triolet 1
12 si quintuplecrochepointee 1

```

## b) Silences

Dans l'ancien format, pour faire un silence, on devait créer une note de volume 0. Dans le nouveau format, il suffit simplement d'écrire le nom du silence sur une ligne. Voici les noms disponibles :

1. **pause** (—). Durée : 4 temps.
2. **demipause** (—). Durée : 2 temps.

3. `soupir (ṭ)`. Durée : 1 temps.
4. `demisoupir (γ)`. Durée :  $\frac{1}{2}$  temps.
5. `quartdesoupir (γ̇)`. Durée :  $\frac{1}{4}$  temps.
6. `huitiemedesoupir (γ̈)`. Durée :  $\frac{1}{8}$  temps.
7. `seiziemedesoupir (γ̈̇)`. Durée :  $\frac{1}{16}$  temps.
8. `trentedeuxiemedesoupir` (j'atteint les limites de mon package de musique en termes de symboles). Durée :  $\frac{1}{32}$  temps.

Les versions pointées sont également disponibles :

9. `pausepointee (⌵)`. Durée : 6 temps.
10. `demipausepointee (⌵)`. Durée : 3 temps.
11. `soupirpointe (ṭ̇)`. Durée :  $\frac{3}{2}$  temps.
12. `demisoupirpointe (γ̇)`. Durée :  $\frac{3}{4}$  temps.
13. `quartdesoupirpointe (γ̇̇)`. Durée :  $\frac{3}{8}$  temps.
14. `huitiemedesoupirpointe (γ̈̇)`. Durée :  $\frac{3}{16}$  temps.
15. `seiziemedesoupirpointe (γ̈̇̇)`. Durée :  $\frac{3}{32}$  temps.
16. `trentedeuxiemedesoupirpointe (\insererIciUnTrenteDeuxiemeDeSoupirPointe{ })`. Durée :  $\frac{3}{64}$  temps.

Les triolets sont encore et toujours disponibles :

17. `trioletdepause`. Durée :  $\frac{8}{3}$  temps.
18. `trioletdedemipause`. Durée :  $\frac{4}{3}$  temps.
19. `trioletdesoupir`. Durée :  $\frac{2}{3}$  temps.
20. `trioletdedemisoupir`. Durée :  $\frac{1}{3}$  temps.
21. `trioletdequartdesoupir`. Durée :  $\frac{1}{6}$  temps.
22. `trioletdehuitiemedesoupir`. Durée :  $\frac{1}{12}$  temps.
23. `trioletdeseiziemedesoupir`. Durée :  $\frac{1}{24}$  temps.
24. `trioletdetrentedeuxiemedesoupir`. Durée :  $\frac{1}{48}$  temps.

Par exemple, les portées suivantes :



seront retranscrites ainsi :

```

1 pause
2
3 do noire 1
4 demipausepointee
5
6 sol# croche 1
7 soupirpointe
8 re trioletdenoire 1
9 trioletdesoupir

```

```

10 mi trioletdenoire 1
11
12 trentedeuxiemedesoupir
13 la-1 quintuplecroche 1
14 seiziemedesoupir
15 si-1 quadruplecroche 1
16 huitiemedesoupir
17 do triplecroche 1
18 quartdesoupir
19 re doublecroche 1
20 demisoupir
21 mi croche 1
22 soupir
23 la noire
24 seiziemedesoupir

```

### c) Commentaires

Dans le nouveau format, il est possible d'insérer des commentaires à l'intérieur des pistes (attention, **il n'est pas possible d'en mettre à l'extérieur**!). Les commentaires commencent par un # suivi d'une espace. Par exemple :

```
1 2 sine
2 # commentaire
3 trioletdedemisoupir # autre commentaire
4 do-1 quadruplecrochepointee # encore un autre commentaire
```

En revanche, ceci n'est pas valide :

```
1 -1
2 tempo 120 # ce commentaire est en-dehors d'une piste
3 1
4 1
5
6 # ce commentaire également
7
8 2 sine
9 # ce commentaire est valide
10 trioletdedemisoupir # celui-ci aussi
11 do-1 quadruplecrochepointee # de même pour celui-là
12
13 # mais pas lui (le programme a lu les deux notes de la piste, donc la piste est
14 # finie à ce stade du fichier)
```

### III.3 Limitations

Cette fonctionnalité supplémentaire souffre de quelques défauts.

a) Autres durées de notes

Entre autres, il est impossible de faire des notes de durée non conventionnelle (comme des notes liées). Par exemple, dans la sonata, on peut lire à la mesure 23, sur la portée de la main droite :



Ici, la liaison de tenue (en rouge) impose une note durant autant qu'une croche pointée et une croche combinées, soit un temps et quart. Cette durée n'étant pas disponible dans le nouveau format, on est obligés de faire deux notes séparées, comme suit :

```

1 sol#1 doublecroche 0.7
2 # ici, la note est censée durer autant que une croche pointée + une croche
3 # (liaison), ce qui est impossible à faire avec mon programme...
4 la1 crochepointee 0.7
5 la1 croche 0.7
6 si1 triplecroche 0.7
7 la1 triplecroche 0.7
8 sol#1 triplecroche 0.7
9 la1 triplecroche 0.7
10 do2 croche 0.7
11 la1 croche 0.7
12 do2 croche 0.7
13 la1 croche 0.7

```

Cela pose un problème lorsque l'on utilise l'ADSR (car la note doublée s'entend). Vous pouvez l'écouter à la 42<sup>e</sup> seconde de la sonata générée depuis `exemples/nouveau_format/sonataadsr.txt`.

## b) Accords

Pour faire des accords, il faut recourir à la méthode des pistes supplémentaires. Par exemple, la portée suivante :



devra être retranscrite comme suit :

```

1 -1
2 tempo 120
3 4
4 1 1 1 1
5
6 4 sine
7 do noire 1
8 do noire 1
9 do noire 1
10 do noire 1
11
12 4 sine
13 soupir
14 mi noire 1
15 mi noire 1
16 mi noire 1
17
18 3 sine
19 demipause
20 sol noire 1
21 sol noire 1
22
23 2 sine
24 demipausepointee
25 do1 noire 1

```

## IV Description de l'ADSR

Ma deuxième fonctionnalité supplémentaire est l'implémentation de l'enveloppe ADSR (Attack, Decay, Sustain, Release). Elle ajoute quatre instruments aux instruments disponibles : `sineadsr`,

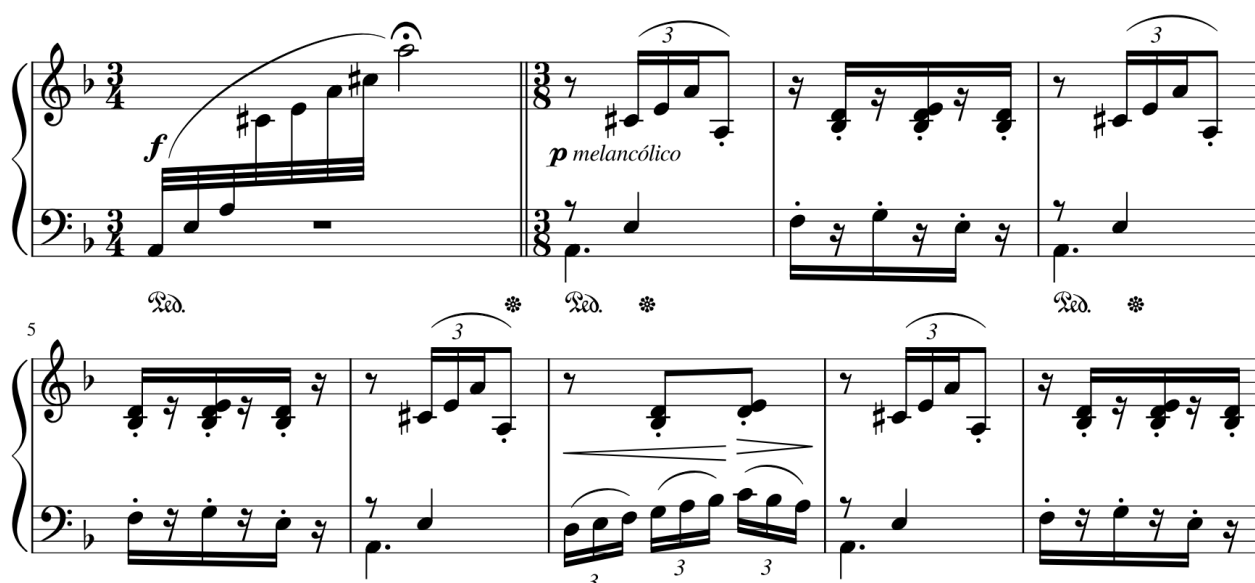
squareadsr, triangleadsr et sawtoothadsr, versions avec enveloppe ADSR de, respectivement, sine, square, triangle et sawtooth.

Cette implémentation de l'enveloppe ADSR est relativement basique. Pendant les 50 premières millisecondes, c'est l'attaque (attack) : l'enveloppe varie entre 0 et 1 de façon linéaire. De la cinquantième milliseconde jusqu'à la centième milliseconde, c'est la chute (decay) : l'enveloppe varie entre 1 et 0,8 de façon linéaire. Ensuite, pendant l'entretien (sustain), l'enveloppe vaut 0,8. À partir d'un certain moment déterminé selon la durée de la note, et jusqu'à la fin de la note, c'est l'extinction (release) : l'enveloppe varie entre 0,8 et 0 de façon linéaire.

Ces valeurs de durée ont été déterminées *a vista de nas*<sup>2</sup>, en fonction de ce qui sonnait bien, mais ça reste à améliorer (je me suis surtout concentré sur le nouveau format de fichier, je n'ai pas pu beaucoup travailler sur l'ADSR).

## V Exemple récapitulatif

À titre d'illustration, voici une petite partition<sup>3</sup> :



Et voici sa retranscription (approximative) :

```

1 -1
2 tempo 100
3 5
4 1 1 1 1 1
5
6 41 triangleadsr
7 la-2 triplecroche 1
8 mi-1 triplecroche 1
9 la-1 triplecroche 1
10 do# triplecroche 1
11 mi triplecroche 1
12 la triplecroche 1
13 do#1 triplecroche 1
14 la1 noire 0.4 # une blanche ne sonnait pas très bien, j'ai préféré mettre une
15 # noire et un soupir
16 soupir
17
18 demisoupir
19 do# trioletdedoublecroche 0.5

```

2. En respectant bien entendu la norme [UNM 00-001](#) de l'Union de Normalisation de la Mécanique.

3. Extrait de Isaac Albéniz, *Rumores de la Caleta*, Op. 71, No. 6, 1887



```

20 mi trioletdedoublecroche 0.5
21 la trioletdedoublecroche 0.5
22 la-1 croche 0.5
23
24 quartdesoupir
25 re doublecroche 0.5
26 quartdesoupir
27 re doublecroche 0.5
28 quartdesoupir
29 re doublecroche 0.5
30
31 demisoupir
32 do# trioletdedoublecroche 0.5
33 mi trioletdedoublecroche 0.5
34 la trioletdedoublecroche 0.5
35 la-1 croche 0.5
36
37 quartdesoupir
38 re doublecroche 0.5
39 quartdesoupir
40 re doublecroche 0.5
41 quartdesoupir
42 re doublecroche 0.5
43
44 demisoupir
45 do# trioletdedoublecroche 0.5
46 mi trioletdedoublecroche 0.5
47 la trioletdedoublecroche 0.5
48 la-1 croche 0.5
49
50 # les notes sont censées être piquées, donc je les raccourcis
51 demisoupir
52 re doublecrochepointee 0.5
53 huitiemedesoupir
54 re doublecrochepointee 0.5
55 huitiemedesoupir
56
57
58
59 21 triangleadsr
60 demisoupirpointe
61 huitiemedesoupir
62 demipause
63 soupirpointe
64
65 quartdesoupir
66 sib-1 doublecroche 0.5
67 quartdesoupir
68 sib-1 doublecroche 0.5
69 quartdesoupir
70 sib-1 doublecroche 0.5
71
72 soupirpointe
73
74 quartdesoupir
75 sib-1 doublecroche 0.5
76 quartdesoupir
77 sib-1 doublecroche 0.5
78 quartdesoupir
79 sib-1 doublecroche 0.5
80
81 soupirpointe
82
83 demisoupir
84 sib-1 croche 0.5

```

```

85 mi croche 0.5
86
87
88
89 13 triangleadsr
90 demisoupirpointe
91 huitiemedesoupir
92 demipause
93 soupirpointe
94
95 demisoupirpointe
96 mi doublecroche 0.5
97 demisoupir
98
99 soupirpointe
100
101 demisoupirpointe
102 mi doublecroche 0.5
103 demisoupir
104
105 soupirpointe
106 soupirpointe
107
108
109
110 27 triangleadsr
111 demisoupirpointe
112 huitiemedesoupir
113 demipause
114
115 la-2 noirepointee 0.5
116
117 fa-1 doublecroche 0.5
118 quartdesoupir
119 sol-1 doublecroche 0.5
120 quartdesoupir
121 mi-1 doublecroche 0.5
122 quartdesoupir
123
124 la-2 noirepointee 0.5
125
126 fa-1 doublecroche 0.5
127 quartdesoupir
128 sol-1 doublecroche 0.5
129 quartdesoupir
130 mi-1 doublecroche 0.5
131 quartdesoupir
132
133 la-2 noirepointee 0.5
134
135 # creshendo
136 re-1 trioletdedoublecroche 0.5
137 mi-1 trioletdedoublecroche 0.6
138 fa-1 trioletdedoublecroche 0.7
139 sol-1 trioletdedoublecroche 0.75
140 la-1 trioletdedoublecroche 0.8
141 sib-1 trioletdedoublecroche 0.9
142 # decreshendo
143 do trioletdedoublecroche 0.8
144 sib-1 trioletdedoublecroche 0.7
145 la-1 trioletdedoublecroche 0.5
146
147
148 11 triangleadsr
149 demisoupirpointe

```

```
150 huitiemedesoupir
151 demipause
152
153 demisoupir
154 mi-1 noire 0.5
155
156 soupirpointe
157
158 demisoupir
159 mi-1 noire 0.5
160
161 soupirpointe
162
163 demisoupir
164 mi-1 noire 0.5
```

## VI Autres exemples

Le dossier `examples/nouveau_format/`, en plus de `albeniz.txt` qui est l'exemple ci-dessus, contient également d'autres exemples : `tests.txt` et `testsadsr.txt` contenant quelques tests basiques, `sbbcfffs.txt` contenant le jingle des CFF (Chemins de Fer Fédéraux suisses), et enfin `sonata.txt` et `sonataadsr.txt`, contenant une version plus longue de la sonate KV545 de Mozart.

## VII Fioritures

Le dossier contient également le fichier `morse_to_dm2.py`, qui convertit une phrase en morse dans (l'ancien) format de fichier (toujours compatible). Trois exemples de phrases se trouvent dans `examples/`, sous les noms `morse.txt`, `morse2.txt` et `morse3.txt`<sup>4</sup>.

---

4. Pour les déchiffrer, il existe des outils en ligne, par exemple [ce site web](#).