

# Projet : Retranscrire une partition musicale depuis un signal sonore.



## Itération 1

Groupe 2 : Carreteros Laetitia , Duraj Bastien , Grolleau Tao , Kempendaers Francis , Jouvét Lucas



# Traitement et analyse du signal sonore

Implémentation d'un algorithme en C permettant de lire un fichier WAVE « commun ».

Recherche d'un signal périodique

Implémentation de la Fast Fourier Transform (FFT)



# Traitement d'un fichier WAVE

- Charge un fichier WAVE en mémoire
- Stocke le temps et l'amplitude associée
- Affiche le signal a l'écran
  
- Ne traite que les fichiers codés sur 16 bits (les plus commun)

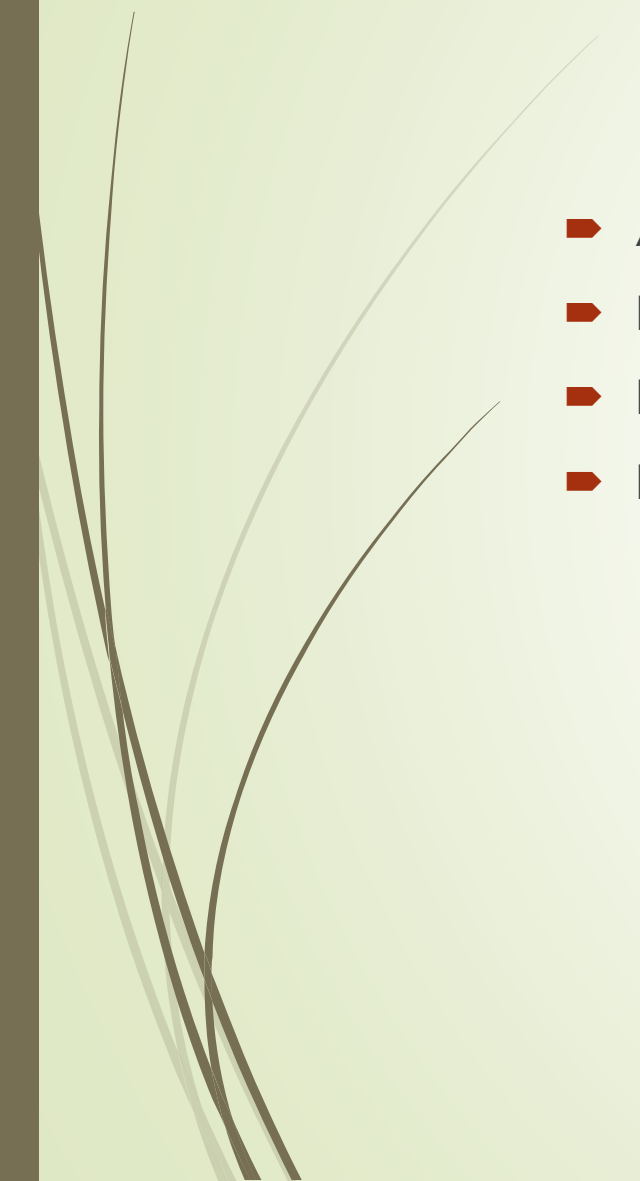


# Recherche d'un signal périodique dans un signal sonore

- Utile pour retrouver une note de musique dans un signal musical propre
- Détection de la période
- Calcul de la fréquence



# Fast Fourier Transform (FFT)

- Algorithme de complexité  $N \log N$  au lieu de  $N^2$
  - $N = 2$  puissance  $k$
  - Permet de récupérer un spectre fréquence/amplitude
  - Permet de récupérer la composition fréquentielle d'un signal périodique
- 



# Problèmes rencontrés lors de l'analyse du signal

- Domaine des mathématiques compliqué
- Utilisation de la théorie des signaux laborieuse
- Beaucoup de choses à apprendre, domaine très vaste



# Traitement d'un fichier midi en un fichier intermédiaire



## Structure du fichier intermédiaire

Nombre de notes

Chiffrage

Composition :  
note octave dièse temps

# Au Clair De La Lune



```
Terminal
File Edit View Search Terminal Help
@@ Other message: class javax.sound.midi.MetaMessage
@@ Other message: class javax.sound.midi.MetaMessage
@@ Other message: class javax.sound.midi.MetaMessage
@@ Channel: 0 Command:176
@@ Channel: 0 Command:192
@@ Channel: 0 Command:176
@@ Channel: 0 Command:176
@@ Channel: 0 Command:176
@@ Channel: 0 Command:176
@@ Other message: class javax.sound.midi.MetaMessage
@@ Channel: 0 Note on, G 04 key=67 velocity: 80
@455 Channel: 0 Note on, G 04 key=67 velocity: 80
@480 Channel: 0 Note on, G 04 key=67 velocity: 80
@935 Channel: 0 Note on, G 04 key=67 velocity: 80
@960 Channel: 0 Note on, G 04 key=67 velocity: 80
@1415 Channel: 0 Note on, G 04 key=67 velocity: 80
@1440 Channel: 0 Note on, A 04 key=69 velocity: 80
@1895 Channel: 0 Note on, A 04 key=69 velocity: 80
@1920 Channel: 0 Note on, B 04 key=71 velocity: 80
@2831 Channel: 0 Note on, B 04 key=71 velocity: 80
@2880 Channel: 0 Note on, A 04 key=69 velocity: 80
@3791 Channel: 0 Note on, A 04 key=69 velocity: 80
@3840 Channel: 0 Note on, G 04 key=67 velocity: 80
@4295 Channel: 0 Note on, G 04 key=67 velocity: 80
@4320 Channel: 0 Note on, B 04 key=71 velocity: 80
@4775 Channel: 0 Note on, B 04 key=71 velocity: 80
@4800 Channel: 0 Note on, A 04 key=69 velocity: 80
@5255 Channel: 0 Note on, A 04 key=69 velocity: 80
@5280 Channel: 0 Note on, A 04 key=69 velocity: 80
@5735 Channel: 0 Note on, A 04 key=69 velocity: 80
@5760 Channel: 0 Note on, A 04 key=69 velocity: 80
@7583 Channel: 0 Note on, A 04 key=69 velocity: 80
@7584 Other message: class javax.sound.midi.Track$ImmutableEndOfTrack
```

```
File Edit Options Buffers Tools Text Help
[Icons] Save Undo [Icons] [Search]
11
4/4
G 0 4 4 0
G 0 4 4 0
G 0 4 4 0
A 0 4 4 0
B 0 4 2 0
A 0 4 2 0
G 0 4 4 0
B 0 4 4 0
A 0 4 4 0
A 0 4 4 0
A 0 4 1 0
```



# Traduction du fichier intermédiaire en partition Lilypond







- On initialise la partition
- Une note en Lilypond : note concaténée à une durée


Exemple :

a8 = La, croche.

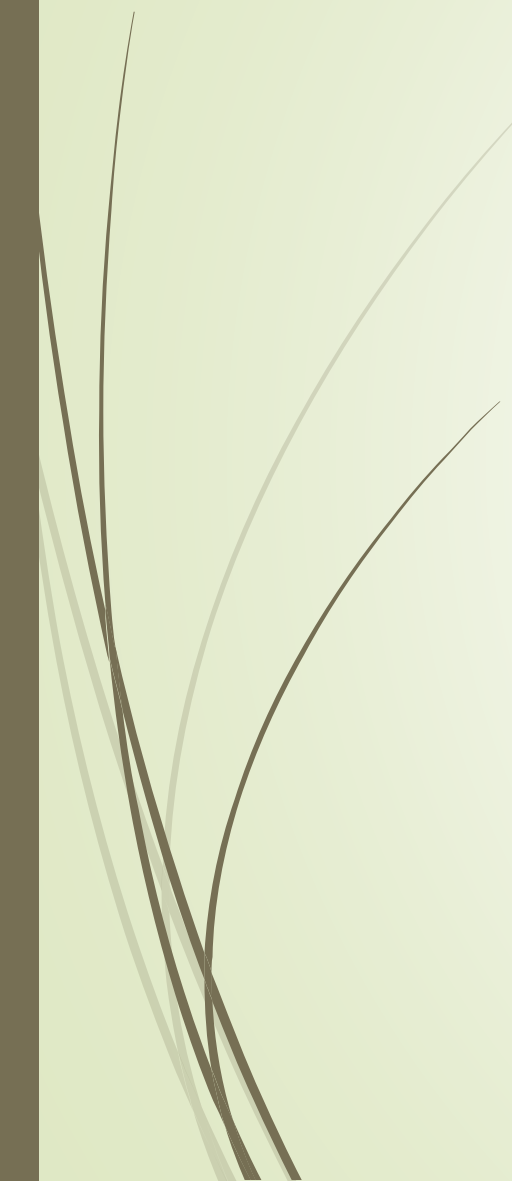
- Pour représenter une liaison entre la note a2 et a4, on écrit : a4( a2)

Démonstration !

	se jouent comme une	
	se jouent comme une	
	se jouent comme une	



# Problèmes rencontrés lors de la traduction en partition

- Longueur de la documentation
  - Comparaison MusicTex/Lilypond
  - Gestion des liaisons de prolongation
- 



# Itération 2

- Pour le traitement du signal :
  - Apprendre à se servir de la FFT
  - Algorithme SFFT pour pouvoir trouver un spectre temps-fréquence
  - Algorithme Passe-Bande pour nettoyer le signal
  - Précision de la détection des périodes et des fréquences
  - Détection de plusieurs notes
- Pour la retranscription en Lilypond :
  - Traitement des valeurs particulières pour les notes
  - Gestion des silences
  - Déterminer l'armure de la partition, ainsi que la hauteur des notes
  - Gérer les répétitions dans une mélodie