

TD séance n° 12

Multimédia Son

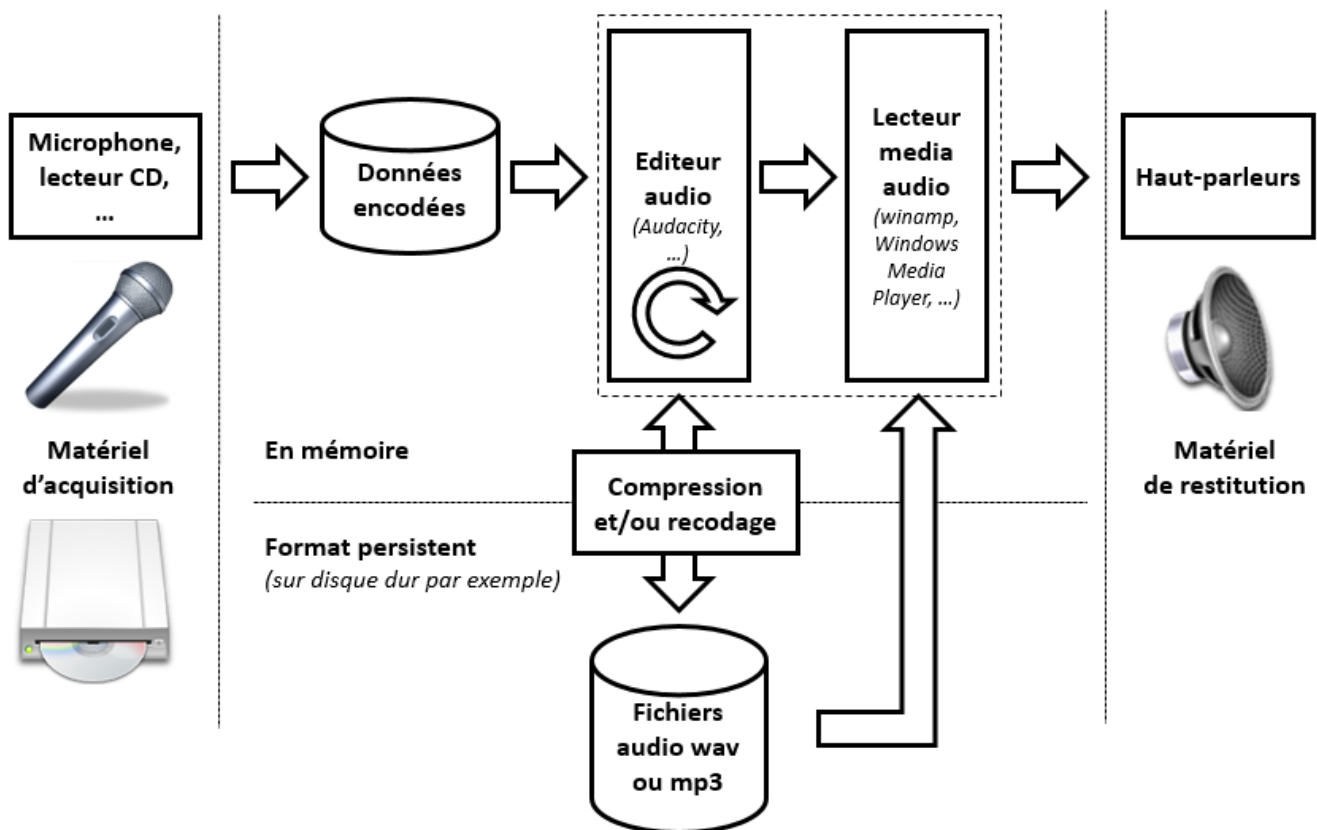
Ce TD est dédié aux chaînes de traitements audio, un des objets multimédias de base. Il se déroulera sous Windows, bien que les logiciels utilisés ou équivalents soient aussi disponibles sous Unix.

Les temps de téléchargement pouvant parfois être long, avant même de débiter le cours, commencez par récupérer les fichiers nécessaires (programmes et fichiers exemples) pour réaliser ce TD (lien ressources sur la page du cours à côté du TD).

La quasi majorité des logiciels que vous utilisez sur votre PC, sur votre box chez vous, sur mobile, sur le Web ..., pour écouter de la musique, travaillent avec les notions que nous allons illustrer ici.

1 Chaîne de traitements pour le son

Une chaîne de traitement pour le son se décline de la même manière que la chaîne de traitement du texte que nous avons étudiée la semaine dernière.



1.1 Mais c'est quoi un son ?¹

Le son est une vibration de l'air, c'est-à-dire une suite de surpressions et de dépressions de l'air par rapport à une moyenne, qui est la pression atmosphérique. D'ailleurs pour s'en convaincre, il suffit de placer un objet bruyant (un réveil par exemple) dans une cloche à vide pour s'apercevoir que l'objet initialement bruyant n'émet plus un seul son dès qu'il n'est plus entouré d'air ! Comme quoi les bruits de combats dans l'espace dépourvu d'air n'est qu'une pure invention d'Hollywood !

Commençons par regarder les extrémités de la chaîne de traitement à savoir l'acquisition et la restitution d'un son.

¹ Source de la partie cours : « Comment ça marche.net »

TD séance n° 12

Multimédia Son

1.2 Acquisition et restitution d'un son

1.2.1 Acquisition

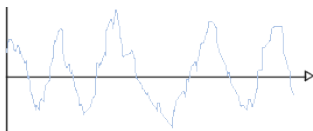
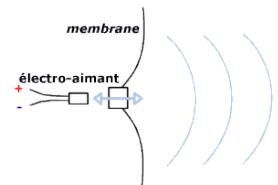
Il y a différentes façons de récupérer un objet multimédia audio. La première, la plus simple, est d'utiliser une source audio déjà sous forme numérique (un fichier audio comme celui stocké sur un CD audio en utilisant un logiciel approprié, un fichier que l'on récupère sur Internet, en ayant vérifié qu'ils sont libres de droits). Ceci est très pratique pour ajouter des bandes sonores à vos objets multimédia composites. Nombreux sont les sites qui vous le permettent.

La deuxième façon consiste à capturer un son du monde réel, grâce à un microphone branché sur la carte audio de votre machine, et de la stocker dans l'ordinateur. Mais comment se déroule ce passage du son physique au son numérique ?

1.2.2 Restitution

La façon la plus simple de reproduire un son actuellement est de faire vibrer un objet. De cette façon un violon émet un son lorsque l'archet fait vibrer ses cordes, un piano émet une note lorsque l'on frappe une touche, car un marteau vient frapper une corde et la fait vibrer.

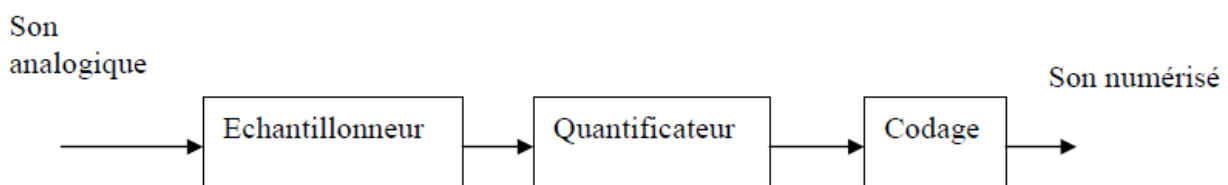
Pour reproduire des sons, on utilise généralement des haut-parleurs. Il s'agit en fait d'une membrane reliée à un électroaimant, qui, suivant les sollicitations d'un courant électrique va aller en avant et en arrière très rapidement, ce qui provoque une vibration de l'air situé devant lui, c'est-à-dire du son !



De cette façon on produit des ondes sonores qui peuvent être représentées sur un graphique comme les variations de la pression de l'air (ou bien de l'électricité dans l'électroaimant) en fonction du temps. On obtient alors une représentation comme montré sur le graphique ci-contre.

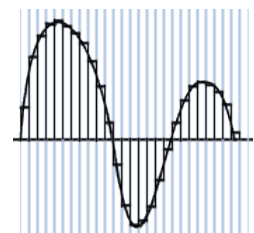
1.3 Représentation du son sous forme numérique

Le passage d'un son analogique (du monde physique) à un son numérisé se déroule suivante 3 étapes que nous allons détailler pour mieux les comprendre.



1.3.1 Echantillonnage et Fréquence

Pour pouvoir représenter un son sur un ordinateur, il faut arriver à le convertir en valeurs numériques ; nous avons vu la semaine précédente que l'ordinateur ne sait travailler que sur ce type de valeurs. Il s'agit donc de relever des échantillons de son (ce qui revient à relever des différences de pression) à des intervalles de temps précis. On appelle cette action l'échantillonnage ou la numérisation du son. L'intervalle de temps entre deux échantillons est appelé taux d'échantillonnage ou la fréquence d'échantillonnage. Le système de conversion numérique du son a été mis au point en 1957 dans les laboratoires Bell par Max Mathews.



TD séance n° 12

Multimédia Son

Voici quelques exemples de fréquences d'échantillonnage et de qualités de son associées :

Taux d'échantillonnage	Qualité du son
44 100 Hz	Qualité CD
22 050 Hz	Qualité radio
8 000 Hz	Qualité téléphone

La valeur du taux d'échantillonnage, pour un CD audio par exemple, n'est pas arbitraire, elle découle en réalité de la **théorie de Shannon**. La représentation numérique fidèle d'un son ne pouvait être obtenue qu'en échantillonnant celui-ci au moins au double de sa fréquence (2, 2 fois pour être précis). Théoriquement, tout échantillonnage numérique devrait donc s'effectuer à 44.1 kHz (44 000 valeurs par seconde), puisque l'oreille humaine peut percevoir les fréquences acoustiques jusqu'à 20.05 kHz.

A chaque échantillon est associée une valeur qui détermine la valeur de la pression de l'air à ce moment, le son n'est donc plus représenté comme une courbe continue présentant des variations mais comme une suite de valeurs pour chaque intervalle de temps.

1.3.2 Quantificateur et codage de l'information (résolution)

Nous avons vu la semaine dernière que l'ordinateur travaille avec des bits pour stocker toute information. Il faut donc déterminer le nombre de valeurs que l'échantillon peut prendre. Cela revient à fixer le nombre de bits sur lequel on code les valeurs des échantillons (donc les surpressions et les dépressions d'air).

- Avec un codage sur 8 bits, on a 2^8 possibilités de valeurs, c'est-à-dire 256 valeurs possibles
- Avec un codage sur 16 bits, on a 2^{16} possibilités de valeurs, c'est-à-dire 65536 valeurs possibles

Avec la seconde représentation, on aura bien évidemment une qualité de son bien meilleure, mais aussi un besoin en mémoire beaucoup plus important. L'inconvénient de ce format (qui est celui utilisé en mémoire) est qu'il n'est pas compact et produit de très gros fichiers.

1.3.3 Plusieurs pistes sons

Enfin, la stéréophonie nécessite deux canaux sur lesquels on enregistre individuellement un son qui sera fourni au haut-parleur de gauche, ainsi qu'un son qui sera diffusé sur celui de droite.

Et bien entendu dans le cas d'un son 5.1, nous aurons 5 pistes pour envoyer le son sur : le haut-parleur avant gauche, avant droit, arrière gauche, arrière droit, celui du centre (ou la voie), ce qui fait 5 canaux plus une dernière piste pour le caisson de basse, ce qui nous donne 5 + 1, noté 5.1.

1.3.4 Conclusion

Un son est donc représenté (numériquement) par plusieurs paramètres :

- la fréquence d'échantillonnage
- le nombre de bits d'un échantillon
- le nombre de voies (une seule correspond à du mono, deux à de la stéréo, etc.)

Il est alors simple de calculer la taille d'une séquence sonore non compressée. En effet, en connaissant le nombre de bits sur lequel est codé un échantillon, on connaît la taille de celui-ci (la taille d'un échantillon est le nombre de bits...).

Pour connaître la taille d'une voie, il suffit de connaître le taux d'échantillonnage, qui va nous permettre de savoir le nombre d'échantillons par seconde, donc la taille qu'occupe une seconde de musique. Celle-ci vaut :

$$\text{Taille Voie} = \text{Taux d'échantillonnage} \times \text{Nombre de bits}$$

Ainsi, pour savoir l'espace mémoire que consomme un extrait sonore de plusieurs secondes, il suffit de multiplier la valeur précédente par le nombre de secondes :

TD séance n° 12

Multimédia Son

Taille Extrait sonore = Taux d'échantillonnage x Nombre de bits x Nombre de secondes

Enfin, la taille finale de l'extrait est à multiplier par le nombre de voies (elle sera alors deux fois plus importante en stéréo qu'en mono...).

La taille en bits d'un extrait sonore est ainsi égale à :

Taille fichier = Taux d'échantillonnage x Nombre de bits x Nombre de secondes x Nombre de voies

1.4 Codage, Décodage, Recodage des données représentant un son

Le codage de chaque échantillon de son, tel que nous venons de le décrire, par « modulation par impulsion et codage » (ou en anglais Pulse Code Modulation) est le format dit **PCM** ; c'est le format brut des données pour encoder un son. Ce format est utilisé dans la phase de numérisation et est aussi utilisé en mémoire pour la manipulation des données par les logiciels. Le traitement **indépendant** de chaque échantillon implique qu'il n'y a ni chiffrement, ni compression des données (nous reviendrons sur ce principe dans le cours final sur les données).

Mais enregistrer directement ces données sur un support de sauvegarde (disque dur ou clé USB), nécessite de grandes capacités de stockage. Pour éviter une perte trop importante de place pour le stockage de ces informations au format brut PCM, plusieurs algorithmes ont été créés. Ces algorithmes sont appelés des **codecs** (compression des mots *coder* et *décoder*). Les plus connus ont donné lieu à des noms qui sont utilisés pour l'extension du fichier qui contient des données compressées avec ces codecs. Il faut alors distinguer deux types d'algorithmes de traitement de ces données brutes : les algorithmes sans perte et avec perte de données.

1.4.1 Codec de compression audio sans perte d'information

Il est donc possible de modifier la manière de représenter l'information à l'aide d'un algorithme qui va encoder et décoder l'information à chaque enregistrement ou relecture des données. Ainsi les données écrites sur le support de stockage seront plus compactes et nécessiteront moins de place sur le support de stockage.

Les algorithmes dits sans perte d'information, sont des algorithmes qui n'altèrent pas (ne dégradent pas) l'information de base (celle qui a été numérisée). L'algorithme se contente de représenter la même information sous une autre forme (avec un autre encodage), cette nouvelle forme occupant moins de place (utilisant moins d'octets) que la forme initiale. Les codecs répondant à cette définition sont peu nombreux :

- [FLAC](#) : Free Lossless Audio Codec (réduction de 30 à 70% par rapport à PCM)
- [ALAC](#) : Apple Lossless Audio Codec (réduction de 40 à 50% par rapport à PCM)
- [Shorten](#) qui signifie raccourcir en anglais.
- et quelques autres moins connus...

1.4.2 Codec de compression audio avec perte d'information

D'autres algorithmes pour la compression et la décompression (codecs) ont été développés. Pour pouvoir compresser encore plus l'information, ces formats sont basés sur le fait de perdre des données qui sont jugées comme non essentielles et permettent ainsi un gain de place important. Cette perte d'information impacte peu la qualité sonore ; on n'entend peu voire pas de différence. Il faut des appareils très performants et une très bonne ouïe pour faire la différence.

Ces codecs avec perte de données sont beaucoup plus nombreux et vous les avez sûrement déjà utilisés :

- [MP3](#) : MPEG-1/2 Audio Layer III (réduction de 80 à 90% par rapport à PCM)
- [AC-3](#) (ou Dolby Digital) : utilisé pour la diffusion TV HD, les supports DVD et Blu-ray
- [Vorbis](#) : plus performant que mp3, mais moins populaire. Souvent utilisé dans les fichiers .ogg.

TD séance n° 12

Multimédia Son

- [WMA](#) : Windows Media Audio (format propriétaire de Microsoft)

Mais sachez toutefois que le fait de compresser les sons dans le cas de la compression avec perte de données, outre le fait de perdre des informations, va changer la dynamique du son. Par exemple, la compression en mp3 consiste à remonter tous les sons faibles, compris entre 5 et 30 décibels (dB), pour en faire des sons plus forts. Cela permet au compresseur d'avoir moins d'amplitude d'information à prendre en compte et donc potentiellement d'utiliser moins de bits pour coder cette information.

Avec un analyseur de niveau sonore (sonomètre), il est possible de voir à quel point ce traitement transforme radicalement les sons (sans en avoir vraiment de perception consciente). Par exemple, en écoutant « Le Boléro » de Maurice Ravel dans un enregistrement sans compression, les courbes qui défilent sur l'écran l'appareil sont d'une grande amplitude. Il y a des silences au début, des notes à peine audibles à 40 dB et puis viennent à la fin les éclats à plus de 85 dB. Donc l'oreille n'est pas tout le temps sollicitée. Il n'y a rien de tout ça sur le même morceau compressé : les courbes démarrent d'emblée à 70-75 dB et se maintiennent à ce niveau pendant toute la durée de l'enregistrement. La dynamique, c'est-à-dire la différence entre un niveau sonore fort et un niveau faible, a disparu. L'oreille est donc plus souvent sollicitée et se fatigue donc plus vite. Vous ajoutez à cela le fait d'écouter avec un casque et, suivant le temps d'écoute, vous avez un cocktail plus ou moins détonnant pour vos oreilles !

1.4.3 Formats de fichiers audios

Attention, certains formats de fichiers audio, identifiables à leur extension, ne correspondent pas forcément au codec qui est utilisé pour le codage de l'information audio. Si un fichier .mp3 va bien contenir des données correspondant à l'encodage de son avec le codec mp3, ce n'est pas toujours le cas.

Certains formats de fichiers peuvent contenir des données encodées avec différents codecs. On appelle ces types de fichiers des **conteneurs** (des fichiers qui peuvent contenir des informations de plusieurs types). Le début du fichier (l'entête) correspond alors à la description de la manière dont les informations suivantes ont été encodées (avec quel codec lire la suite des données).

C'est le cas par exemple des fichiers .wav, .aif ou .ogg qui peuvent contenir des informations dans différents formats.

1.5 Lecteurs Audio

La plupart des lecteurs logiciels audio (*players*) lisent la quasi-totalité des formats. Certains, comme Windows Media Player, chargent les codecs² dynamiquement pour de nouveaux formats. Il n'en va pas de même des lecteurs intégrés à du matériel dit de salon ou vos téléphones mobiles. Les lecteurs typiques et les extensions des fichiers de base associés sont :

- RealOne Player: .ra
- Windows Media Player: .wma
- Winamp : .mp3, .ogg
- ou un lecteur générique comme VLC qui permet de lire à la fois l'audio et la vidéo pour de nombreux codecs.

Vous pouvez utiliser charger le lecteur logiciel qui vous conviendra le mieux à partir du moment où il contient le codec permettant de décoder les informations de son.

² Codec = « codage-décodage ». C'est un algorithme permettant d'encoder ou décoder un flux ou un signal. Un codec peut être un circuit électronique, intégré ou, dans le cas qui nous intéresse, un morceau de programme.

TD séance n° 12 Multimédia Son

1.6 Dispositif de restitution (CD audio)

Les données gravées sur un CD audio sont des données non compressées (format `.pcm`, utilisé dans les fichiers `.wav` fournis dans le TD), avec une fréquence d'échantillonnage de 44.100 Hz. Nombreux sont les logiciels de gravure qui peuvent vous être fournis avec un graveur de CD audio, comme Néro, BurnAtOnce, Express Burn, ...

2 Installation des logiciels

Nous utiliserons le magnétophone de Windows pour effectuer les compressions/recodages (ex. réglages de l'échantillonnage, compression, ...) et Audacity pour étudier/manipuler le signal audio (amplitude, spectre en fréquence, ...).

Pour installer les logiciels dont vous aurez besoin pour ce TD, vous pouvez utiliser le lien suivant :

<http://stephane.lavirotte.com/teach/cours/envinfo1/12%20TD%20Multimedia%20Son.zip>

2.1 Audacity

Audacity est un logiciel libre pour la manipulation de données audio numériques. Audacity permet d'enregistrer du son numérique par le biais d'une entrée audio sur une carte son. Il permet d'appliquer divers filtres et effets : pitch, tempo, réduction de bruit, égaliseur, filtres de Fourier, augmentation de fréquences précises, compression, amplification, normalisation, écho, phaser, wahwah, inversion...

2.2 Magnétophone Windows

Pour ouvrir les fichiers audios, nous utiliserons une application fournie sur Windows : le magnétophone de Windows (dans Programmes / Accessoires / Divertissement, ...). Cette application n'est plus présente sous Windows 7. Nous utiliserons donc le programme `sndrec32.exe` que vous trouverez dans l'archive.

Pendant que vous téléchargez et installez les logiciels nécessaires à ce TD ainsi que les exemples sur lesquels nous allons travailler, nous allons commencer par un petit calcul théorique.

TD séance n° 12

Multimédia Son

Durant les exercices qui suivent, nous allons tenter de comprendre par la pratique le codage du son, le recodage et la compression dans différents formats ainsi que les traitements que l'on peut effectuer une fois le son numérisé.

Pour débiter ce TD, vous commencerez par télécharger le fichier de ressources et installerez les logiciels Audacity, et les librairies `ffmpeg` et `lame` qui sont fournis dans l'archive que vous avez téléchargée.

3 Exercices

3.1 Audio numérique

Exercice n°1:

Quelle sera la taille d'un fichier audio avec les paramètres suivants : stéréo, d'une durée d'une heure, encodé à 44 100Hz, en 16 bits ? Convertir la valeur trouvée en bits en K_{io} puis M_{io} . A quel support de stockage correspond la capacité de stockage en M_{io} que vous venez de trouver ?

Exercice n°2:

Nous travaillerons maintenant travailler sur des exemples avec les fichiers `music1.wav` et `music2.wav`. Vous utiliserez le logiciel `sndrec32.exe` qui se trouve dans le fichier de ressources fourni avec le sujet de TD).

Quelles sont les caractéristiques de numérisation (menu Fichier / Propriétés) de ces deux sons : durée, fréquence d'échantillonnage, nombre de bits de quantification, nombre de canaux, format ?

Quelle est la taille de chacun de ces fichiers en octets et M_{io} ? (Clic droit sur le fichier dans l'explorateur, Propriétés)

Comment expliquez-vous la différence de taille entre ces deux fichiers pour à peine 1 seconde de différence ?

Exercice n°3:

Nous allons maintenant faire quelques manipulations qui vont nous permettre d'illustrer les variations de qualité de l'objet multimédia audio que nous pourrions obtenir selon les propriétés des dispositifs d'acquisition et des codages de base. Il s'agit, par exemple, de comparer le format 8 bits et 16 bits en utilisant le fichier `music1.wav`. D'origine la bande est enregistrée en 16 bits. Pour la convertir en 8 bits (sans changer les autres paramètres), vous pouvez utiliser le magnétophone de Windows : `sndrec32.exe` (menu Fichier / Propriétés / Convertir...).

A partir du fichier `music1.wav`, convertissez-le en 8 bits, et sauvegardez le sous `music1_8bits.wav`.

Écoutez les deux fichiers obtenus durant l'exercice précédent. Que se passe-t-il ? Quelles sont les tailles des fichiers `music1.wav` et `music1_8bits.wav` ? Pourquoi ?

Exercice n°4:

Continuons en comparant des sons de plus faible intensité. Avec le logiciel Audacity, reprenez le fichier `music1.wav`, vous l'atténuez de 30dB (Effets / Amplification / -30dB). Enregistrez le (menu Fichier / Exporter l'audio) sous `music1_attenué.wav`. Puis convertissez-le (comme vous venez de le faire avec `sndrec32.exe`) au format 8 bits, et sauvegardez le sous `music1_8bits_attenué.wav`.

Écoutez les deux fichiers obtenus. Pourquoi êtes-vous obligés d'augmenter le volume ? Que constatez-vous au niveau de la qualité du son atténué par rapport au non atténué ?

Exercice n°5:

A l'aide de `sndrec32.exe`, vous allez reprendre le fichier d'origine `music1.wav`. Vous allez convertir le fichier d'origine en qualité radio stéréo (`music1_radio.wav`) et en qualité téléphonique (`music1_tel.wav`).

Pour chacun des deux formats : que pensez-vous de la qualité du signal ? Que pouvez-vous en conclure ?

TD séance n° 12 Multimédia Son

3.2 Formats de compression de sons

Dans Audacity, grâce au menu « Fichier / Exporter l'audio », vous pouvez ré-encoder les données du son dans différents codecs : flac, mp3, vorbis, wma, ...

Nous allons maintenant comparer les conséquences du ré-encodage des fichiers audio avec différents codecs audios.

Exercice n°6:

Compresserez le fichier `music2.wav` dans chacun des formats de compression ci-dessus. Puis dans un tableau, vous donnerez vos conclusions en ce qui concerne la qualité des sons compressés obtenus et la taille des fichiers.

Exercice n°7:

Que pouvez-vous conclure sur les performances de chacun de ces codecs audios ?

Exercice n°8:

D'après les tailles que vous obtenez pour chacun des fichiers, proposés dans ranger chacun d'eux dans la catégorie des codecs avec perte et sans perte de données.

3.3 Edition et manipulation de sons numériques

Audacity permet aussi de faire des manipulations sur le son numérique. Le logiciel possède des fonctions avancées pour ceux qui sont habitués à l'édition audio : montage, mixage, conception, échantillonnage, enregistrement, sound design, mastering ... D'autres logiciels gratuits pour l'édition audio existent comme le logiciel Wavosaur (<http://fr.wavosaur.com/>).

Exercice n°9:

Donnez le nom des opérations à effectuer si l'on désire :

- Supprimer les silences de début et de fin
- Atténuer le son en début et en fin de lecture
- Générer un morceau de musique en lecture inverse (les plus mélomanes reconnaîtront une manipulation à l'origine du « backmasking », notamment utilisé par un groupe de musique célèbre, lequel ? wikipedia est mon ami ...)

Vous devez maintenant avoir compris la manière de représenter un son numériquement, l'impact des différents paramètres sur la qualité du son et les différents formats d'encodage du son pour l'enregistrement.

Le reste est une affaire de curiosité et de temps...

4 Synthèse sur le codage de l'information

Exercice n°10:

Si on vous donne le codage binaire suivant : 00000000 01000000, pouvez dire quelle est le type d'information représentée ou quel est le type de média auquel vous avez à faire ? Quelles sont les valeurs entières représentées par chacun de ces octets en décimal et en hexadécimal ?

Exercice n°11:

Dire ce que les données binaires précédentes représentent dans les cas suivants :

- un texte encodé en ANSI
- un texte encodé en UTF-16
- un son encodé en 8 bits

TD séance n° 12 Multimédia Son

- un son encodé en 16 bits
- une image en niveaux de gris
- une image couleur avec une palette (on considère qu'on retrouve ces valeurs pour des points de l'image)
- une image en vraie couleurs (on considère que ce sont les 2 premiers octets d'une couleur)