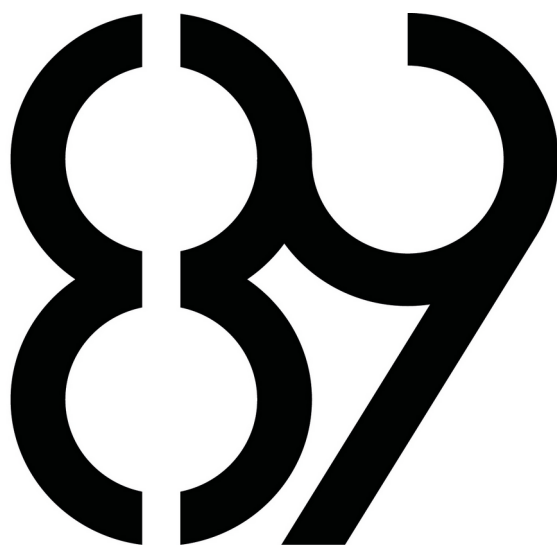


LAPINS NOIRS

Programmation audio



## SAMPLEUR

Composer avec un son existant

- Le Laboratoire aux Lapins Noirs -  
[pedagogie@ecole-89.com](mailto:pedagogie@ecole-89.com)

*Le « sampling » est né de la possibilité d'enregistrer sur un support électronique un son et ensuite d'en accélérer ou ralentir la lecture, de l'inverse, de l'amplifier, de le filtrer... Transformant chaque son potentiellement en instrument de musique.*

*La musique concrète est le premier genre musical à exploiter cette possibilité en usant des sons exclusivement de bruits de la vie courante :*

<https://www.youtube.com/watch?v=c4ea0sBrw6M>

*Cela a motivé ensuite la création de machine générant du son, les fameux synthétiseur.*

*Ce document est strictement personnel et ne doit en aucun cas être diffusé.*



Programmation audio

# INDEX

- 01 – Accéder à la mémoire audio
- 02 - Sampler
- 03 - Enveloppe



## 01 – Accéder à la mémoire audio

Pour créer un espace dans la carte son, utilisez la fonction **bunny\_new\_effect** qui prend en paramètre une durée du son en seconde. Cette durée est un flottant, cela signifie qu'il est possible demander un nombre de seconde non rond.

La mémoire renvoyée, de type **t\_bunny\_effect** contient de nombreux champs dont le plus intéressant est **sample**. Ce champ est similaire au champ **pixels** du **t\_bunny\_pixelarray** dans le sens où c'est **la donnée qui sera exploitée**. La longueur de l'espace mémoire pointée par **sample** est **duration** multiplié par **sample\_per\_second**.

Si **duration** n'a certainement aucun secret pour vous : c'est le paramètre passé à **bunny\_new\_effect**. Le champ **sample\_per\_second** est par contre probablement inconnu : il s'agit de la fréquence d'échantillonnage, c'est à dire le nombre de niveau que l'onde sonore peut prendre en une seconde. Plus cette valeur est élevée, plus le son est de bonne qualité et des nuances peuvent être entendues.

Par défaut, la fréquence d'échantillonnage est 44100Hz, c'est à dire qu'il faut écrire dans 44100 cases de **sample** pour faire un son d'une seule seconde... Chaque niveau enregistrée l'est dans un entier de 16 bits, c'est à dire un entier dont les valeurs possibles sont comprises entre -32768 et 32767. Plus l'amplitude – la valeur absolue des valeurs formant l'onde sonore – est grande, plus le volume est important.

Vous allez commencer simplement en réalisant une **friture**, de la même manière que celle que vous deviez réaliser **graphiquement** : remplissez **sample** de valeurs aléatoires à l'aide de la fonction **rand**. N'hésitez pas à tenter d'exploiter toute l'amplitude disponible, entre -32768 et 32767, mais *faites attention à mettre le volume à un niveau raisonnable avant de tester*.

```
void e89_set_noise(t_bunny_effect *fx);
```

Pour jouer le son, une fois que vous l'avez écrit, vous devrez d'abord appeler **bunny\_compute\_effect** qui permet de faire passer le son que vous avez écrit à la carte son. Ensuite, vous pourrez appeler **bunny\_sound\_play**. Vous remarquerez que **bunny\_sound\_play** prend un pointeur sur **t\_bunny\_sound** et non un pointeur sur **t\_bunny\_effect**... mais ce n'est pas grave, car il y a un **t\_bunny\_sound** dans **t\_bunny\_effect** ! Envoyez simplement son adresse et préparez vous à danser avec modération.



## 02 – Sampler

Utilisez `bunny_load_effect` et `bunny_new_effect` pour respectivement charger des sons et créer une musique. Les sons à charger devraient contenir tous la même note, pourquoi pas un unique LA international à 220Hz ?

Programmez les fonctions suivantes :

```
double e89_get_note_frequency(const char *note, int octave);

void e89_write_sample(t_bunny_effect *dest, t_bunny_effect *src, int start, int stop, double freq, bool loop);
```

La fonction `get_note_frequency` renvoi la fréquence de la note indiqué dans `note` à l'octave `octave`. Vous trouverez les informations concernant les fréquences sur Wikipedia :

[https://fr.wikipedia.org/wiki/Note\\_de\\_musique#Fr%C3%A9quence\\_d'une\\_note](https://fr.wikipedia.org/wiki/Note_de_musique#Fr%C3%A9quence_d'une_note)

Seul les octaves 1, 2 et 3 sont demandés, mais vous pouvez également fournir tous les autres de -1 à 9. Pour cela, renseignez vous sur la logique présente derrière ces fréquences, remarquez que le LA évolue de manière intéressante d'un octave à l'autre : 55, 110, 220, 440, 880, etc.

<https://www.youtube.com/watch?v=cTYvCpLRwao>

La fonction `write_sample` écrit dans `dest` l'onde présente dans `src` de `start` à `stop` à la fréquence `freq` sachant que le son dans `src` est à 220Hz. Si le son est plus court que la distance entre `start` et `stop`, il n'est joué qu'une fois, sauf si `loop` est vrai, dans ce cas, on boucle.

La fonction `write_sample` *ajoute* au son existant et n'écrase pas.

Pour rappel :

Qu'est ce que cette fréquence signifie ? La fréquence d'un son détermine sa note : plus la fréquence est élevée, plus la note sera aiguë. Une fréquence de 100 indique que en **une seconde, il y a 100 ondulations**. L'acuité humaine est généralement comprise entre 20Hz et 20 000Hz, en dessous et au-delà, il est rare de parvenir à entendre le son.



## 03 – Enveloppe

Une enveloppe peut-être assignée à n'importe quel son. Ajoutez une enveloppe simple, comme sin, à votre son lorsqu'il est ajouté.