Rapport TP : MUNUM

Synthèse sonore musicale avec FAUST

Viozelange Matis

date : 15 octobre 2024

École Centrale Nantes

PARTIE 1 : Synthèse additive

1.1 Contrôle du volume et de la fréquence

Code pour régler fréquence et amplitude :

```
import("stdfaust.lib");
process = os.osc(frequency) * gain;
gain = hslider("gain", 0.1, 0, 1, 0.01);
frequency = hslider("freq", 440, 0, 20000, 1);
```

1.2 Envelope temporelle

```
gate = button("gate");
profile = en.ar(0.002, 0.1);
envelope = gate : profile;
process = os.osc(440) * envelope;
```

La ligne profile du code ci-dessus crée un profil d'attaque de 0.002s et de relâchement de 0.1s. L'enveloppe associe le profil au bouton *gate*.

En contrôlant les gains du profil avec le code ci-dessous, on peut obtenir un son percussif avec un petit temps d'attaque et un son avec du *sustain* en allongeant l'attaque et le relâchement.

```
gate = button("gate");
profile = en.ar(0.002 * attack, 0.1 * release);
envelope = gate : profile;

attack = hslider("attack", 1, 0, 100, 0.1);
release = hslider("release", 1, 0, 10, 0.01);

process = os.osc(440) * envelope;
```

1.3 Harmonic additive model

On analyse l'influence des harmoniques de ce code :

```
import("stdfaust.lib");

partialslider(n) = hslider("partial %n", 0.25, 0, 1, 0.01);
partial(n,f) = os.osc(f*n) * partialslider(n);
freqslider = hslider("freq", 200, 100, 800, 1);
process = sum(i, 4, partial(i+1, freqslider));
```

On remarque que la hauteur d'une note n'est pas influencée par l'absence du fondamental. Cependant, si on atténue les fréquences impaires (1 et 3), la note semble monter d'une octave.

Augmenter n ne change pas la hauteur de la note mais son timbre. De manière générale, jouer sur les harmoniques change d'abord le timbre d'une note comme l'a si bien remarqué Rousseau.

PARTIE 2 : Synthèse soustractive

2.1 Écho

```
bounce = @(4410 * slider) : *(0.75);
slider = hslider("gain", 1, 0, 10, 0.02);
echo = +~bounce;
process = echo;
```

Le code ci-dessus crée un effet d'écho sur un signal d'entrée. Le terme *@(4410 * slider)* sert à choisir la fréquence de l'écho en jouant sur le retard des signaux en écho.

L'opérateur " " sert à donner un feedback à l'entrée de la boucle. Le terme de retour est simplement le signal de sortie, ce qui crée cet effet d'écho.

2.2 Algorithme de Karplus-Strong

On implémente le code ci-dessous :

```
import("stdfaust.lib");
mean(x) = (x+x')/2;
delayslider = hslider("delay", 0, 0, 200, 1);
gainslider = hslider("gain", 0, -0.98, 0.98, 0.01);
feedback = @(delayslider) : mean : *(gainslider);
noiseslider = hslider("noise", 0.5, 0, 1, 0.01);
process = no.noise * noiseslider : + ~ feedback;
```

La fonction *mean* joue le rôle d'un filtre passe-bas en faisant la moyenne entre x et x', où x' est x pris à un instant précédent et proche. Le feedback permet de distribuer l'énergie du bruit sur les fréquences cibles pour synthétiser le son voulu. Plus le gain tend vers 1, plus le son passe du bruit à un signal harmonique. Le délai quant à lui joue sur la hauteur du son perçu.

2.3 Synthèse d'une guitare

Pour la synthèse de la guitare, on ajoute une enveloppe avec le code vu ci-dessus pour le réalisme et on pousse le gain très proche de 1. On obtient les réglages figure 2.1.

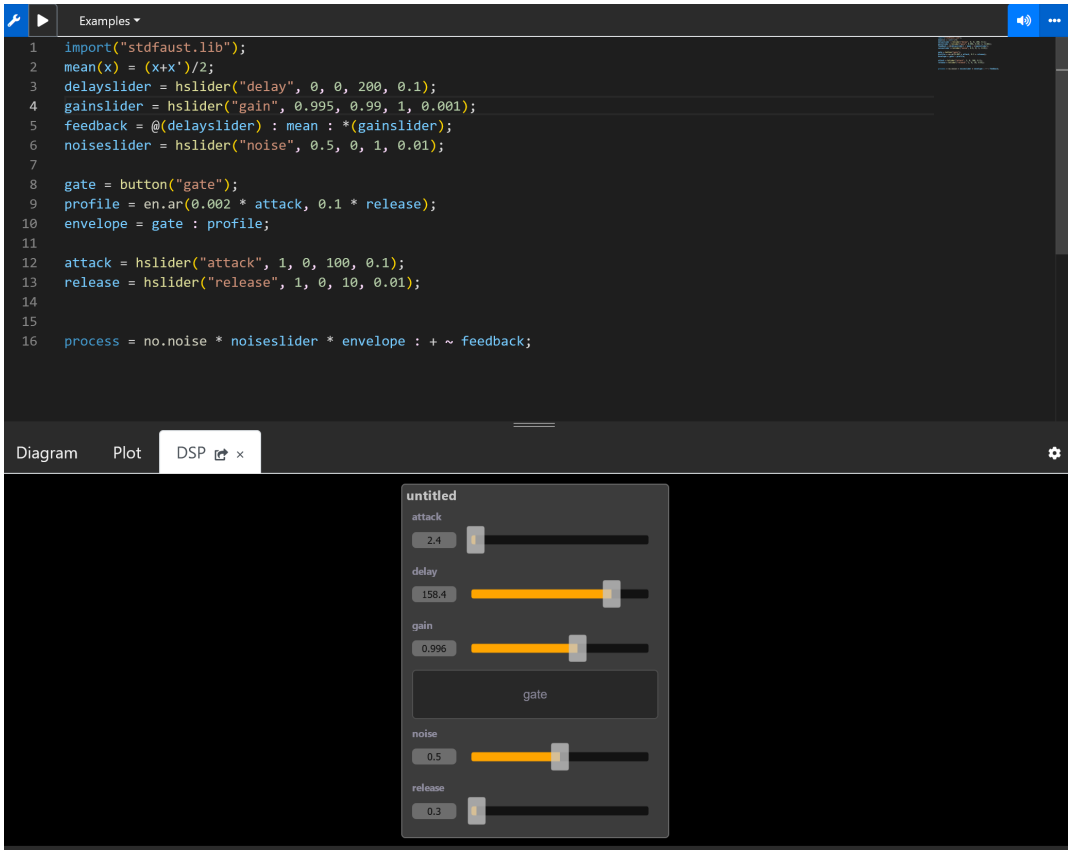

FIGURE 2.1 – Réglages pour la synthèse d'une guitare