

Workshop Faust à Improtech/Uzeste 2023

Objectif

L'objectif de ce workshop est de vous initier à Faust, un langage de programmation permettant de créer des instruments de musique électroniques. Le workshop s'adresse à tous et ne nécessite pas de connaissances préalables en programmation.

Durant le workshop, nous allons utiliser un outil en ligne, le Faust IDE, qui va nous permettre d'écrire et de faire fonctionner nos programmes Faust directement dans un navigateur Web. Pour accéder à cet outil, il suffit d'ouvrir la page <https://faustide.grame.fr> depuis son navigateur.

Grâce à cet outil, nous allons créer, étape par étape, un petit synthétiseur musicale que l'on pourra ensuite jouer soit directement depuis l'ordinateur, soit à partir d'un clavier MIDI, soit également depuis un smartphone.

Nous terminerons le workshop par une petite improvisation musicale collective, en utilisant l'instrument que nous aurons construit.

Etape 1

Baisser le volume; être en mode MONO !

```
import("stdfaust.lib");  
process = os.sawtooth(50);
```

Essayer →

Etape 2

Ajout d'un contrôle de volume :

```
import("stdfaust.lib");  
process = os.sawtooth(50)  
    * hslider("gain", 0.1, 0, 1, 0.01);
```

Essayer →

Etape 3

Ajout d'un contrôle de fréquence :

```
import("stdfaust.lib");  
process = os.sawtooth(hslider("freq", 50, 30, 8000, 1))  
    * hslider("gain", 0.1, 0, 1, 0.01);
```

Essayer →

Etape 4

Ajout d'un filtre résonnant :

```
import("stdfaust.lib");
process = os.sawtooth(hslider("freq", 50, 30, 8000, 1))
    * hslider("gain", 0.1, 0, 1, 0.01) :
fi.resonlp(300,5,1);
```

Essayer →

Etape 5

Restructuration du code :

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

freq = hslider("freq", 50, 30, 8000, 1);
gain = hslider("gain", 0.1, 0, 1, 0.01);

process = os.sawtooth(freq)*gain
    : fi.resonlp(300,5,1);
```

Essayer →

Etape 6

Oscillateur sur la fréquence de résonance :

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

freq = hslider("freq", 50, 30, 8000, 1);
gain = hslider("gain", 0.1, 0, 1, 0.01);
lfo = os.lf_triangle(0.5)*0.5+0.5;

process = os.sawtooth(freq)*gain
    : fi.resonlp(lfo*300+50,5,1);
```

Essayer →

Etape 7

Oscillateur sur la fréquence de résonance, ajout de contrôles sur le lfo :

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

freq = hslider("freq", 50, 30, 8000, 1);
gain = hslider("gain", 0.1, 0, 1, 0.01);
lfreq = hslider("lfreq",0.5,0.01,50,0.01);
lrange = hslider("lrange",300,10,5000,0.01);
```

```
lfo = os.lf_triangle(lfreq)*0.5+0.5;

process = os.sawtooth(freq)*gain
        : fi.resonlp(lfo*lrangle+50,5,1);

Essayer →
```

Etape 8

Production d'un son stéréo :

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

freq = hslider("freq", 50, 30, 500, 1);
gain = hslider("gain", 0.1, 0, 1, 0.01);

lfreq = hslider("lfreq",0.5,0.01,50,0.01);
lrangle = hslider("lrangle",300,10,5000,0.01);
lfo1 = os.lf_triangle(lfreq)*0.5+0.5;
lfo2 = os.lf_triangle(lfreq*1.01)*0.5+0.5;

process = os.sawtooth(freq)*gain
        <: fi.resonlp(lfo1*lrangle+50,5,1),
           fi.resonlp(lfo2*lrangle+50,5,1);

Essayer →
```

Etape 9

Pilotage à partir d'un clavier MIDI (Poly 16) :

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

freq = hslider("freq", 50, 30, 500, 1);
gain = hslider("gain", 0.1, 0, 1, 0.01);
gate = button("gate");

lfreq = hslider("lfreq",0.5,0.01,50,0.01);
lrangle = hslider("lrangle",300,10,5000,0.01);
lfo1 = os.lf_triangle(lfreq)*0.5+0.5;
lfo2 = os.lf_triangle(lfreq*1.01)*0.5+0.5;

process = os.sawtooth(freq)*gain*gate
        <: fi.resonlp(lfo1*lrangle+50,5,1),
           fi.resonlp(lfo2*lrangle+50,5,1);

Essayer →
```

Etape 10

Ajout d'un limiteur pour éviter les clics :

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

freq = hslider("freq", 50, 30, 500, 1);
gain = hslider("gain", 0.1, 0, 1, 0.01);
gate = button("gate");

lfreq = hslider("lfreq", 0.5, 0.01, 50, 0.01);
lrange = hslider("lrange", 300, 10, 5000, 0.01);
lfo1 = os.lf_triangle(lfreq)*0.5+0.5;
lfo2 = os.lf_triangle(lfreq*1.01)*0.5+0.5;

process = os.sawtooth(freq)*gain*gate
  <: fi.resonlp(lfo1*lrange+50,5,1),
     fi.resonlp(lfo2*lrange+50,5,1);

effect = co.limiter_1176_R4_stereo;

Essayer →
```

Etape 11

Ajout de contrôles MIDI :

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

freq = hslider("freq", 50, 30, 500, 1);
gain = hslider("gain", 0.1, 0, 1, 0.01);
gate = button("gate");

lfreq = hslider("lfreq", 0.5, 0.01, 50, 0.01);
lrange = hslider("lrange[midi:ctrl 49]", 300, 10, 5000, 0.01);
lfo1 = os.lf_triangle(lfreq)*0.5+0.5;
lfo2 = os.lf_triangle(lfreq*1.01)*0.5+0.5;

process = os.sawtooth(freq)*gain*gate
  <: fi.resonlp(lfo1*lrange+50,5,1),
     fi.resonlp(lfo2*lrange+50,5,1);

effect = co.limiter_1176_R4_stereo;

Essayer →
```

Etape 12

Ajout d'une enveloppe et d'un volume :

```
import("stdfaust.lib");
freq = hslider("freq", 50, 30, 500, 1);
gain = hslider("gain", 0.1, 0, 1, 0.01);
gate = button("gate") : en.adsr(0.1, 0.2, 0.8, 2);
volume = hslider("volume[midi:ctrl 48]",0.6,0,1,0.01);

lfreq = hslider("lfreq",0.5,0.01,50,0.01);
lrange = hslider("lrange[midi:ctrl 49]",300,10,5000,0.01);
lfo1 = os.lf_triangle(lfreq)*0.5+0.5;
lfo2 = os.lf_triangle(lfreq*1.01)*0.5+0.5;

process = os.sawtooth(freq)*gain*gate
  <:  fi.resonlp(lfo1*lrange+50,5,1),
      fi.resonlp(lfo2*lrange+50,5,1);
effect = co.limiter_1176_R4_stereo;

Essayer →
```

Etape 13 (Keyboard version)

Ajout d'un écho, la vitesse contrôle en partie le lrange :

```
declare name "Wahoo";

import("stdfaust.lib");
freq = hslider("freq", 50, 30, 500, 1);
gain = hslider("gain", 0.1, 0, 1, 0.01);
gate = button("gate") : en.adsr(0.1, 0.2, 0.8, 2) ;
volume = hslider("volume",0.6,0,1,0.01);

lfreq = hslider("lfreq[midi:ctrl 48]",0.5,0.01,4,0.01);
lrange = hslider("lrange[midi:ctrl 49]",300,20,5000,0.01) * (1+2*gain);
lfo1 = os.lf_triangle(lfreq)*0.5+0.5;
lfo2 = os.lf_triangle(lfreq*1.01)*0.5+0.5;

process = os.sawtooth(freq)*gain*gate
  <:  fi.resonlp(lfo1*lrange+50,5,1)*volume,
      fi.resonlp(lfo2*lrange+50,5,1)*volume;

effect = par(i,2,ef.echo(1,0.25, 0.75))
  : co.limiter_1176_R4_stereo;

Essayer →
```

Etape 14 (Android version)

Metadata acc ajoutées aux lfo :

```
declare name "Wahoo";

// version pour android
import("stdfaust.lib");
freq = hslider("freq", 50, 30, 500, 1);
gain = hslider("gain", 0.1, 0, 1, 0.01);
gate = button("gate") : en.adsr(0.1, 0.2, 0.8, 2) ;
volume = hslider("volume",0.6,0,1,0.01);

lfreq = hslider("lfreq[acc: 0 3 -10 0 10]",0.5,0.01,4,0.01);
lrange = hslider("lrange[acc: 1 3 -10 0 10]",300,10,5000,0.01) * (1+gain);
lfo1 = os.lf_triangle(lfreq)*0.5+0.5;
lfo2 = os.lf_triangle(lfreq*1.01)*0.5+0.5;

process = os.sawtooth(freq)*gain*gate
  <: fi.resonlp(lfo1*lrange+50,5,1)*volume,
    fi.resonlp(lfo2*lrange+50,5,1)*volume
  : effect;

effect = par(i,2,ef.echo(1,0.25, 0.75))
  : co.limiter_1176_R4_stereo;
```

Essayer →

Installer l'application sur Android

L'application pour Android est disponible ici ou via le QR code ci-dessous :

Installer l'application sur iOS

L'application native n'est pas disponible pour iOS. Il faut donc utiliser la version Web préalablement créée. On peut l'ouvrir à partir du QR code suivant :