LINFO1104
MaestrOZ project report

Alexandre Flamant 5308 1500
See full source

April 12, 2022

1 Known issues and limitations

After multiple testing, I came to the conclusion that the program met the intended behaviour without limitations.

2 Non declarative parts of the program

Given the hypothesis that all functions on Mozart/OZ List documentation page are declarative then all the program is declarative. The only blur exception to this is the extensive use of the *List.make* function and for loop to set list elements. The documentation states:

> "Returns a list of length I. All elements *are fresh variables*."

Under the hypothsis that, List.forAll and List.forAllInd are implemented recursively, the program only declare variables and assign value once. So it is declarative.

3 Surprising implementations

I intensively use the *List.make* function in combination with the List.forAll and List.forAllInd functions.

Here is an snippet from the program:

```
fun {NoteSample Note}
  Sample = {List.make {Float.toInt {Float.round SamplingSize * Note.duration}}}
  NoteFrequency = {Frequency Note}
in
  {List.forAllInd
      Sample
      proc{$ I Ai}
         Ai = 0.5 * {Float.sin (2.0 * Pi * NoteFrequency * {Int.toFloat I}/SamplingSize)}
      end}
  if Smoothing then
     local DT = {Min 0.015 0.2 * Note.duration} in {Fade DT DT Sample} end
  else
     Sample
  end
end
```

I use a procedure to set the values of the list. Without declaring any recursive function or Cell counter. It results, in my opinion, in a clearer and shorter code which proved easier to debug.

4 Extensions to the program

4.1 Smoothing of the notes

This extension is activated by setting the *Smoothing* variable to *true* in the Control Variable section at the top of the code.

This is done in the *NoteSample* function by using the *Fade* function used for the fade filter. As showed in the snippet. The enveloppe I used for the smoothing is a trapezoid one with 15ms fade on both ends of the sound

4.2 Additionnal filter

All .dj.oz test files and the resulting .wav files are available in the /sample folder of the source present in the source.

4.2.1 Siren filter

The siren filter mimic the effect of an alarm on a sound. It is declared using the following record structure:

siren(minf:<Float> maxf:<Float> spike:<Float> <Music>

Where:

- minf is the minimum intensity factor of the alarm
- maxf is the maximum intensity factor of the alarm
- spike is the number of high note in the the whole music

The applied envelope make sharp loud note and smooth low note. To do so the envelope F is:

$$F_i = -(f_{max} - f_{min}) \mid \cos\frac{i\pi S}{L-1} + \pi \mid + (f_{max} - f_{min}) + f_{min}$$

with $f_m in =$ minf $f_m ax =$ maxf $S =$ spike and L the length of the sample list.

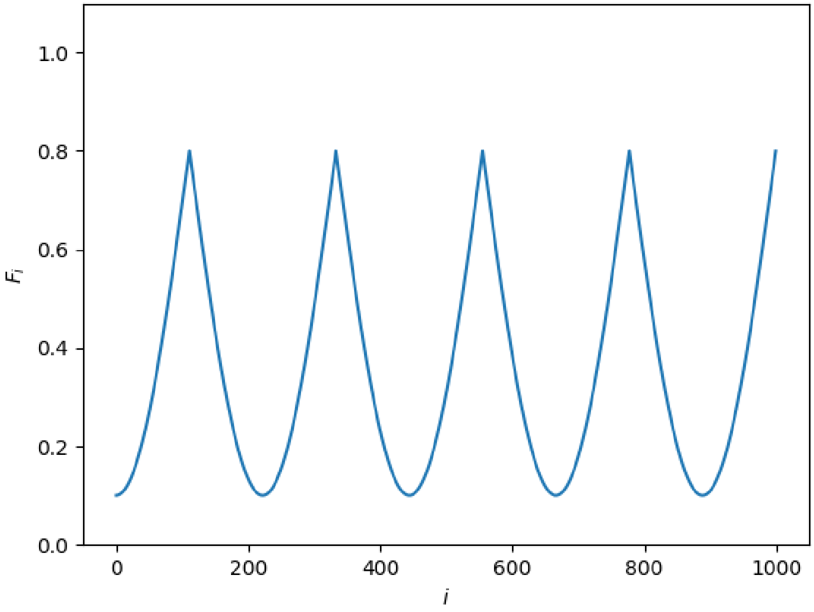

Figure 1: Envelope of siren(minf:0.1 maxf:0.8 spike:4.5 <Music>)

4.2.2 Cross fade filter

The cross fade filter make a smooth transition from a music to another by fading out the first one while the other starts. It is declared using the following record structure:

crossfade(seconds:<Float> <Music>$_1$ <Music>$_2$)

Where:

- seconds is the transition time between the musics
- <Music>$_1$ is the first music
- <Music>$_2$ is the second music

The CrossFade function uses the fade function on both music to fade the end of <Music>$_1$ and the start of <Music>$_2$. Silence is added to <Music>$_2$ in order to place it to the right position compared to <Music>$_1$. The result signal is the sum of these sample.

4.2.3 Vibrato filter

The vibrato filter make the sound vibrate similarily to vibrato with an instrument. It is declared using the following record structure:

vibrato(frequency:<Float> decay:<Float> <Music>

Where:

- frequency is the frequency of the vibrato
- decay is the variation of intensity between loud and low notes of the vibrato

The applied envelope use sharp loud note and smooth low note. To do so the envelope F is:

$$F_i = \frac{d}{2}\cos i2\pi f + (f_{max} - f_{min}) + (1 - \frac{d}{2})$$

with d = decay and f = frequency.

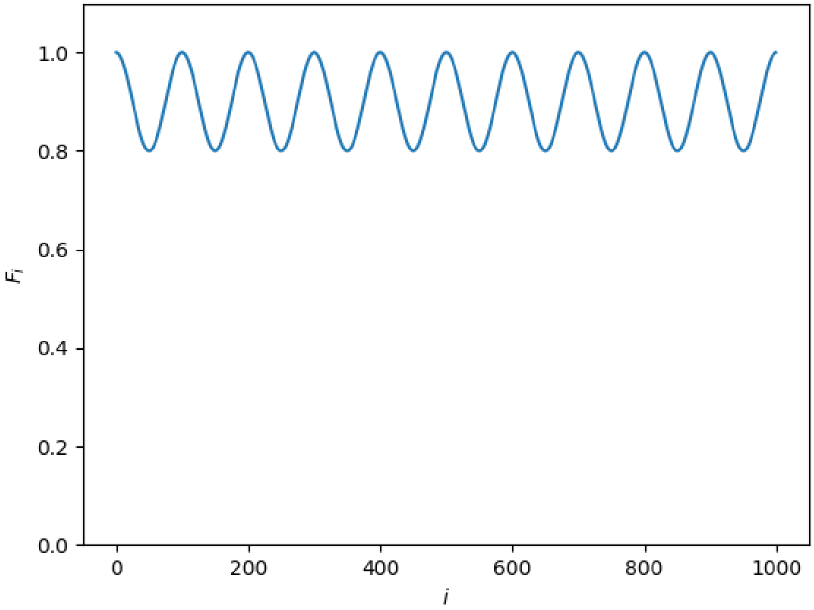

Figure 2: Envelope of vibrato(decay:0.2 frequency:0.01 <Music>)