Elody Sample Document

Using the TimeLine Editor

D. Fober - S. Letz - Y. Orlarey
(fober,letz,orlarey)@rd.grame.fr

In this example we'll mainly see how to do temporal applications using the TimeLine Editor. These applications will specify the rhythm, the nuance and transpositions of the melodic pattern showed in figure 1.

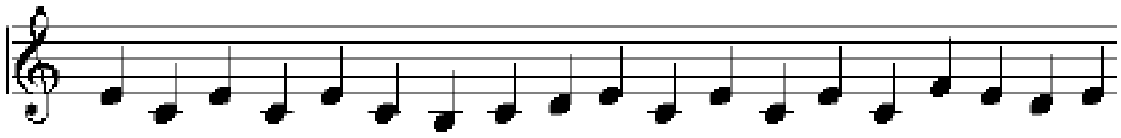

figure 1

MELODIC PATTERN AND FUNCTIONS

The expression *melody* correspond to the melodic pattern in figure 1. It's obtained by simply playing the melody using the Keyboard constructor. The transformations applied to melody should produce the result showed in figure 2.

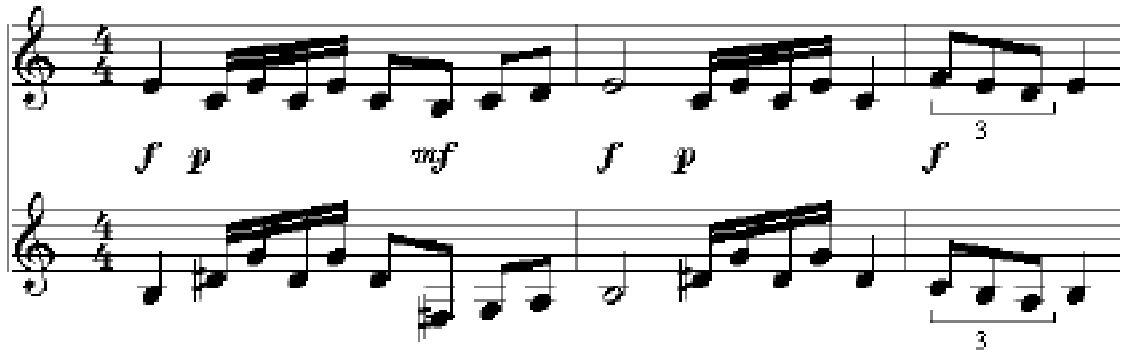

figure 2

Rhythm functions
To express the rhythm, we'll build several functions corresponding to different note durations: *fHalf*, *fQuarter*, *fEight* and *fSixteenth*. Applied to a basic Elody note, they will stretch them to the relative durations of a half note, quarter, eight and sixteenth notes.
The function *fToQuarter* stretch several notes to the duration of a quarter note. It is used to specify non regular beat divisions like triplets for example.

Nuance functions
To express the nuances, we need 3 functions named *p, mf* and *f*. They produces velocity transpositions.

Transposition functions
To express the second voice, we need 2 transposition functions :

fTransp-5 : transpose its argument 5 1/2 tons down.
fTransp+3 : transpose its argument 3 1/2 tons up.

FUNCTIONS IN TIME SPACE

We'll now put the functions in time space using the TimeLine Editor. Open a new TimeLine Editor and drag the unnamed expression to the main box of the editor (the little box at the left of the time scale). You should get a result like in figure 3.

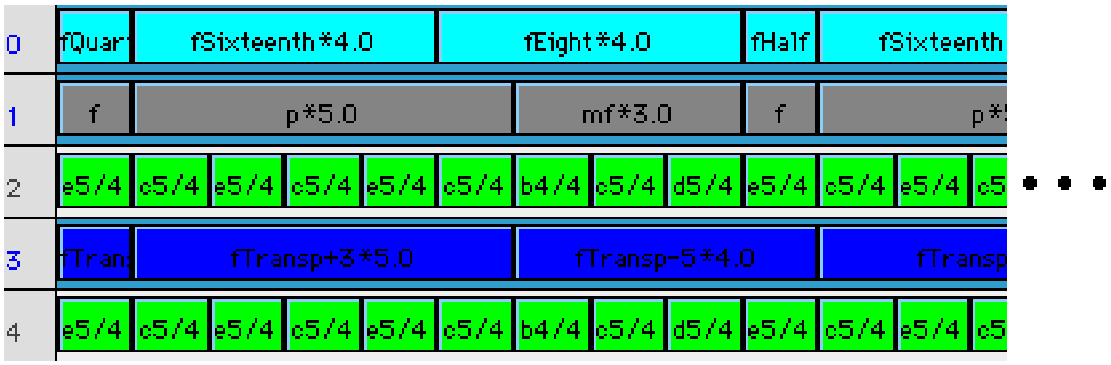

figure 3

• Tracks 2 and 4 are normal tracks, they contain the previously defined melodic pattern.
• Track 3 is a function track, it is applied to the track 4 and contains a sequence of transposition functions. These functions are stretched to the desired duration in order to be applied to the corresponding sections of the melodic pattern. The result of Track3 applied to Track4 is then mixed with the Track 2.
• Track 1 is a function track. Like the Track 3, it contains a sequence of functions stretched to be applied to specific parts of the melodic pattern. These functions are nuance functions. They are applied to the previous result (ie Track2 mixed with the result of Track3 applied to Track4).
• Track 0 is also a function track. It specifies the rhythm of all the previous result. Once again, it contains a sequence of rhythm functions stretched to different durations.

Note that the rhythm functions are applied last. Because they modify the duration of their arguments, once applied the graphic representation cannot be used to synchronize different expressions with their results.