Elody Sample Document

A Basic Tonal Approach

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

In this example we'll see a simple way to express tonal music. In fact, there are several possible approaches, they are more related to music theory than to particular programming technics. Each one of them has specific expressive capabilities. Although limited, the following example shows a simple way to include tonal capabilities in an Elody expression.

BUILDING SCALES AND DEGREES

First of all, we build a scale: expression *baseScale* in the document, in fact a C Major scale. We simply used the Keyboard constructor to build it. Then, we'll cut all the different degrees in this scale using any appropriate Elody constructor: begin and rest operators, or Begin-End constructor

For example:
let *base* be the duration of the different notes of my scale (in fact a basic Elody note)
the different degrees (C, D, E, F...) may be expressed as:

```
C := begin (baseScale, base)
D := begin (rest(baseScale,base), base)
E := begin (rest(rest(baseScale,base),base), base)
etc...
```

where `begin` and `rest` are the corresponding Elody operators.

Because every Elody expression contains its history, the defined degrees contain now the baseScale which may be made variable. Abstracting this baseScale from a degree only keep the process of cutting at the corresponding degree location within this scale. Applying this abstraction to a different scale will apply the cutting process to the new scale and will produce a new note with the same tonal status within this scale.
Note that it will only work with scales where notes have the same duration.

TONAL TRANSPOSITIONS

We use now the previous degrees to build a simple melody (figure 1), stored in the *melody* expression.

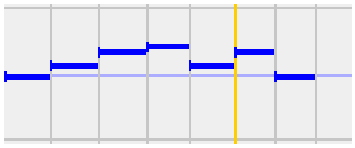

figure 1

It is the sequence of the following degrees : C, D, E, F, D, E, C ie:

```
melody := seq (C, D, E, F, D, E, C)
```

In this melody we'll then make the base scale variable:

```
fMelody := lambda (baseScale, melody)
```

Applied to different scales, the resulting *fMelody* function will then produces the corresponding tonal transpositions.
Applied to a descending C Major scale and because of the cutting process (see upward), the function will produce the mirror of the melodic pattern.