EXPLODE: a user interface for sequencing and score following.

Miller Puckette
IRCAM, 31 Rue St. Merri, 75004 Paris, France

A graphical sequence editor, EXPLODE, is described, aimed at editing lists of objects with a uniform data structure with several numeric parameters. EXPLODE offers some control over which parameters are to be viewed, and how they will affect the appearance of an event on the screen. Applications to score following are described, which were cumbersome to set up before EXPLODE was written, and which take effective advantage of EXPLODE's viewing flexibility.

Why would anyone want to write another graphical sequence editor? The research literature is full of them (the ideas represented in the SCRIVA editor [1] come closest to what is described here) and software companies are competing to develop the ultimate commercial model. But the following issues, which have come up constantly in IRCAM's music production environment, have not been solved:

- Real-time sequencers, particularly those with graphical editing, are mostly MIDI-based. Computer Music tends not to fit easily into seven bits.
- Much more flexibility is needed in how the data that makes up a sequence is "played". The ideal is to regard the sequence as a mere data structure and to traverse it in any way desired for playback.
- Heavy use is made at IRCAM of score following, in which the computer is expected to synchronize itself with a live player. The graphical editor should make it easy to specify this.
- The editor should interface with the MAX program [2] for the MacIntosh, which is used for most real-time computer music here.

The EXPLODE sequence editor attempts to fill these needs. EXPLODE is implemented as a MAX "external object"; i.e. it is not part of the MAX program itself but is loaded dynamically by MAX when invoked. EXPLODE is still a prototype, but has already been used once in concert, in a fifty-minute piece for orchestra and live electronics.

THE EDITOR.

An EXPLODE window appears in Figure 1: a control panel and a collection of events. The controls are, from left to right, the drawing mode, the play/record/follow indicators, three switches, and the "parameter box". The parameter box contains cells for showing/editing the parameters' numerical values, and switches for choosing which parameter to show as "y" (the vertical location of the event), "w" (the horizontal width) and "#" (a number printed above the event.) In Figure 1 these are the defaults: pitch, duration, and nothing, respectively; other possibilities arise below.

The three modes allow you to move events (changing either their start times or their pitches), to stretch them (dragging horizontally or vertically changes the "w" and "#" value, respectively) and to make new events by clicking and stretching. The indicators blacken when the sequence is being used through its MAX interface (see below). Of the three switches, the most interesting one is "show ranges"; this toggles the parameter box into showing the scales of the parameters. The pitch, for example, will be accurate to a cent if its scale is reduced by 100. The "60" in the left margin indicates that middle C is currently the (MIDI standard) 60: after reducing by 100 this would be 6000. The "y" parameter also gets a choice of grid type; it can be the staves shown here, or a numerical grid (more appropriate if "y" is showing a non-pitch value.)

When a selection is made, the parameter box prints the common parameters if the selected events, or "*" if they differ among the selection. Typing into the parameter box sets the corresponding parameter for all selected events. For instance, to change an arpeggio into a chord, select all its events and type a number into the "time" box.

THE MAX INTERFACE.

EXPLODE appears as a box in MAX as shown in Figure 2. (It is invoked here as "exp2", a test version.) The five inlets and outlets correspond to the five parameter fields: time, pitch, velocity, duration, and channel. For recording, EXPLODE takes sets of five parameters and inserts them as events; the events can be generated in any way that can be programmed in MAX. For playing, the "next" method causes the next event to be output; it is up to the surrounding patch to decide when to send "next" messages and what to do with the outputs.

For score following, incoming events are sent to EXPLODE as in recording, and conditionally generate events on the output. EXPLODE regards channel 1 as the set of events to follow, and all other channels as an accompaniment. Incoming events which match an event on channel 1 cause the matched event to be output. The matched events need not come out in exactly the same order as they occur in the sequence; moreover, if an event is skipped in the input it never appears on the output. Events whose channels are not 1 are output as soon as they are passed; i.e. as soon as a later event from channel 1 is matched.

Using the matched notes of channel 1 in this way, a live score can be "embellished" with hidden parameters. Figure 3 shows a simple example, in which the notes of an A major scale are to be mapped onto different MIDI channels. The channels (encoded as velocities in EXPLODE) are printed above the notes (the (vel, #) square is dark in the parameter box). MAX can use the performed velocities and durations, so these fields can be used arbitrarily by EXPLODE in this context.

Figure 4 shows the use of channel 2 to provide an accompaniment to a simple melody. Channel is being shown as the width of the notes; the scale is channel 1 and the "accompaniment" is channel 2. The velocities of the accompaniment are almost always used as cues; here cue number 1 happens on A, 2 on E, and 3 on the second A. For legibility (and to show off another feature) the staves have been suppressed.

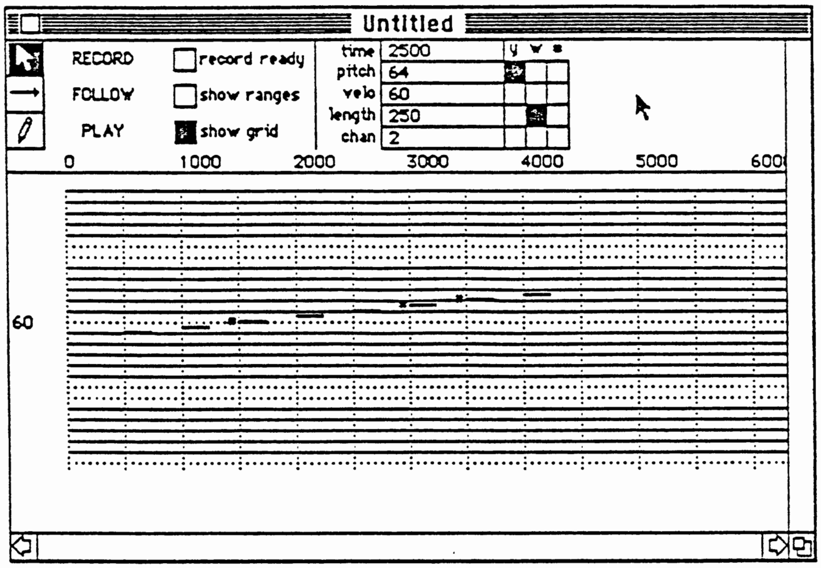
WHAT NOW?

The reader will note that the parameter names (time, pitch, velocity, duration. channel) smack of MIDI. Work is currently under way to make this list extensible; this is added justification for the parameter box, which will become indispensable as the number of parameters rises.

REFERENCES.

[1] Buxton, William, et al, "The evolution of the SSSP score-editing tools." Foundations of Computer Music, (Roades and Strawn, editors), MIT press, 1985.

[2] Miller Puckette, "The Patcher", Proceedings, ICMC 1988 (Cologne).



time 2500
pitch 64
velo 60
length 250
chan 2

☐ record ready
☐ show ranges
■ show grid

Figure 1. An EXPLODE window.

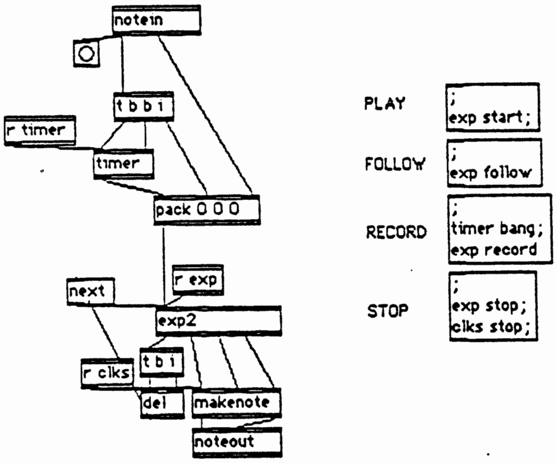


Figure 2. EXPLODE's MAX interface.

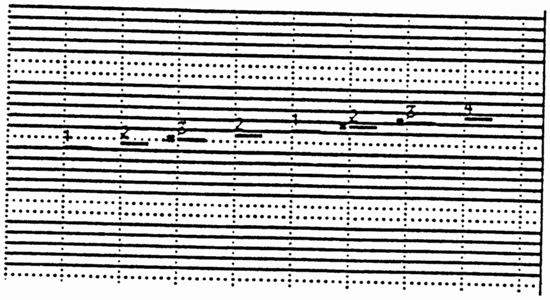


Figure 3. Adding hidden parameters to a score.

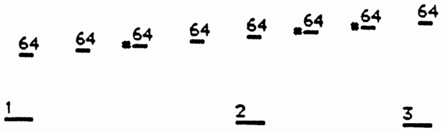


Figure 4. Using another channel for score-following accompaniment.