

note~ for Max - An extension for Max/MSP for Media Arts & music

Thomas Resch
University of Music Basel, FHNW
Leonhardsstrasse 6
4053 Basel Switzerland
admin@noteformax.net

ABSTRACT

note~ for Max consists of four objects for the Software Max/MSP which allow sequencing in floating point resolution and provide a Graphical User Interface and a Scripting Interface for generating events within a timeline. Due to the complete integration into Max/MSP it is possible to control almost every type of client like another software, audio and video or extern hardware by note~ or control note~ itself by other software and hardware.

Keywords

Max/MSP, composing, timeline, GUI, sequencing, score, notation.

1. INTRODUCTION

The four objects were developed in the C-programming language and provide a GUI, a scripting interface, and a timeline for recording, editing, and playing back control data. A new data format has been developed, which allows storing floating point lists of arbitrary length plus text within one event, an approach Miller Puckette already proposed with his Max object "explode" in 1990 [1]. Of course, compatibility to MIDI is still guaranteed.

2. THE OBJECTS

2.1 The note~ object

The Region Editor is integrated into the main note~ object and looks like the arrange window of common sequencing software. Regions are created either with the mouse or by sending the message `newRegion[regionname][track][timestamp][duration]` to note~:

```
newRegion A_NEW_REGION 0 1. 16.
```

Using attributes, the behavior and appearance of note~ can be customized. note~ sends and receives a sync signal, hence it is possible to synchronize note~ to the Max transport and to Max for Live or synchronize note~ objects among themselves, which enables the user to create complex polyrhythmic structures very easily by using different BPM for every note~ object. Through the Status Information Outlet, note~ provides feedback for GUI- and scripting-interaction and therefore its functionality can be extended by common Max/MSP objects.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

NIME'13, May 27-30, 2013, KAIST, Daejeon, Korea.
Copyright remains with the author(s).

By using `regionStart` and `regionEnd` Status Information and the provided sync signal it is possible, for example, to control/playback audio and video files and place them within the timeline. In playback mode, note~ sends the event data from the sequencer data outlet as a list consisting of the track number, event type (which basically replaces MIDI channel and controller number), and a floating point list of arbitrary length plus text. Storing additional data besides the usual four MIDI-parameters in one event allows for real polyphonic control of complex instruments. Enabling the attribute `outputNextEvent` forces note~ to output not only the current but also the next event which makes interpolation of every kind between event parameters possible.

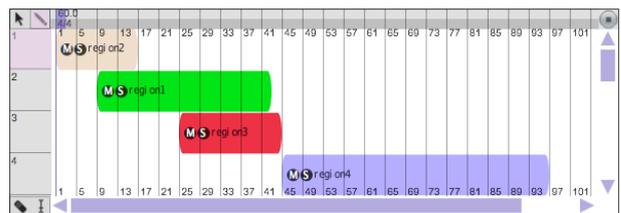


Figure 1. The note~ object

2.2 The note.eventEditor object

Here, events can be created with the mouse or by sending the message `newEvent[regionname][eventtype][timestamp][pitch][duration][velocity][.]` to note~:

```
newEvent A_NEW_REGION 0 1. 60.5 1.5 80.7 1. 3. 5.3 6.7  
127. 3000. 5000. 139.2 "Many many numbers and a lot of text"
```

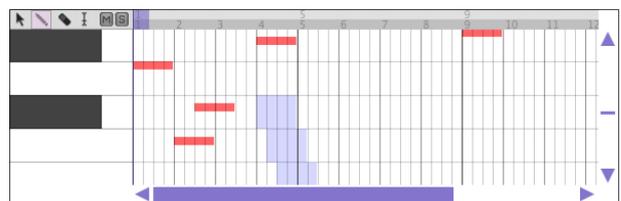


Figure 2. The note.eventEditor object

MIDI-Channels, -Control and -Note Messages are replaced by freely assignable "event types" which can undertake every kind of functionality. A double click on an event opens a small text editor, the parameter editor, where the whole parameter list can be edited. Every parameter of an event has floating point resolution. Therefore, it is possible to create microtonal pitches with a 32 Bit floating point accuracy.¹ The data backend is

¹ With the recent transition of Max/MSP to 64 Bit floating point resolution, note~ will also have 64 Bit accuracy in the near future.

integrated into the note~ object. Therefore, all messages, even the event-related ones like newEvent() or selectEvent() have to be sent to the bound note~ instance. The currently active and editable event type is displayed opaque while all other event types appear transparent. By sending the message groupEvent[regionname][keyword] it is possible to bind events together for GUI editing:

groupEvent A_NEW_REGION selected

A possible scenario for grouping events could be to automatically create whole overtone spectra on every mouse down within the Event Editor. This could be done by sending note~s newEvent status information to a buffer where the spectral data is stored. So every time the user creates an event with the mouse, the original pitch and velocity is sent to the buffer and gets multiplied by the stored relative spectrum. Every overtone generates a newEvent() message with the keyword *select* which creates already selected events:

newEvent select [original event type] [original timestamp] [overtone pitch] [original duration] [overtone velocity]

Then the groupEvent() message is sent. The user is able to create a couple of hundred events on every mouse down and has the ability to edit those event groups together with the mouse.

2.3 The note.score object

The note.score object provides classic western music notation, extended for the needs of contemporary music including microtonality and nested triplets. The score layout is accessible either through a context menu or also by sending messages, for example:

selectEvent A_NEW_REGION dsharp || pitch >= 80.
setAccidental A_NEW_REGION selected flat
splitEvent A_NEW_REGION selected 3

These messages will select all notes with either the pitch dsharp or a pitch greater then 80 (in MIDI Pitch), then replace all sharps of these selected notes with flats and finally split all notes into three, meaning for example that a quarter note will become a triplet.

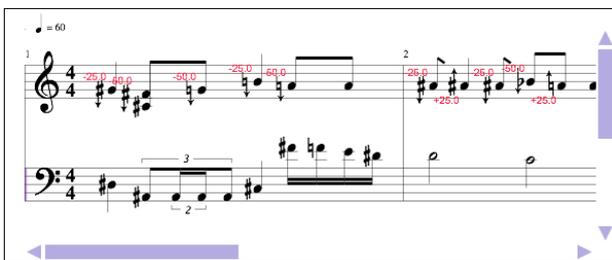


Figure 3. The note.score object

Complex rhythmical structures can also be created by sending a newEvent() message with additional split information:

newEvent A_NEW_REGION 0 1. 60.5 1. "3(1 3 2(1 3))" 80.

The 1. before the quotation mark represents the overall duration, a quarter note, which is then split into a triplet, where the first note stays as it is, the second one is split into three and

the last one split into two notes. The first note of these two stays as it is and the second one is split again into another three notes. Another application of adding split information to the newEvent() message would be to create many events at once by splitting a very long duration:

newEvent A_NEW_REGION 0 1. 60.5 400. "1200" 80.

In this case the duration 400. is split by 1200 which would create 1200 events with the duration 0.25, therefore 1200 1/16 notes.

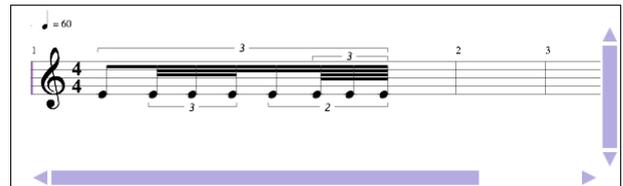


Figure 4. Nested triplets in note.score

2.4 The note.time object

The note.time object translates from seconds to beats², BarBeats³, hh:mm:ss and vice versa.

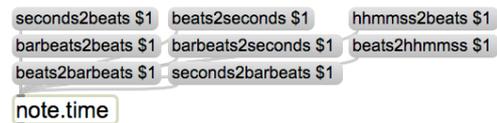


Figure 5. The note.time object

3. SCRIPTING

All of note~s functionality is accessible through Max-Messages in a database-like syntax:

selectEvent A_NEW_REGION all
editEvent A_NEW_REGION selected pitch += 0.5⁴
deleteEvent A_NEW_REGION selected
deleteRegion A_NEW_REGION

This series of function calls selects all events of the region "A_NEW_REGION", then adds 0.5 to every pitch of every selected event, then deletes all selected events, and finally deletes the region itself.

4. DATA FORMAT

A note~ score is saved as a .txt file in plain text and consists of Max-Messages which are readable by humans. Therefore translation and parsing of note~ scores is easily done also outside a note~/MaxMSP environment. It is also possible to save a note~ score automatically with the Max Patcher containing the note~ object.

5. CONCLUSION

From the beginning of note~s development, an emphasis has been placed on achieving an aesthetic neutrality, meaning that

² One beat represents a quarter note, independent from time signature

³ BarBeats describe time in the form bar x beat y, dependent from time signature

⁴ 0.5 corresponds to a quarter tone

note~ should not influence the artist's style of composing. Therefore, the arrangement and Event Editor view of note~ have been designed to be as familiar and user-friendly as possible in order to prevent an artistic affection. note~ allows the creation of complex scores for (contemporary) music and Media Arts in addition to realtime data generation, manipulation and playback in concert- and installation-situations. By saving its data in plain text, note~ enables reconstruction outside Max/MSP and note~, thereby fulfilling an important basic requirement for longterm archiving[2].

Although there is already software out there which allows linear graphical editing (every commercial sequencer), manipulating events and creating and displaying complex rhythmical structures (Open Music, Super Collider), whenever several of those disciplines shall be combined, in most cases there is no other possibility but to export or rather degrade data to the MIDI format in order to continue working with another application. This might either limit the working process from the beginning due to the anticipation of data loss or there will be data loss. note~ tries to resolve this issue by providing a lot of functionality of the above mentioned disciplines and of

course by integrating itself into the Max/MSP programming environment and therefore allowing the user to add and extend all functionality note~ itself lacks.

6. ACKNOWLEDGMENTS

Thanks to Prof. Erik Oña, PhD and Dr. Michael Kunkel for their support and supervision of the project.

7. REFERENCES

- [1] Miller Puckette, ICMC Proceedings 1990, pp. 346-352, ISBN:1-58113-345-6, 1990.
- [2] Raymond A. Lorie, JCDL'01 Proceedings, pp. 346-352, ISBN:1-58113-345-6, 2001.

8. Appendices

For detailed information about Max/MSP please refer to www.cycling74.com.

At www.noteformax.net note~ is freely available as download including documentation, reference and video tutorials.