

connected to it, which are the rendered score elements.

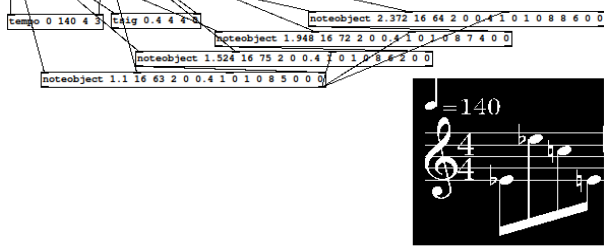


Figure 2: Abstractions are automatically connected together (top-left) to create a beamed group (bottom-right)

A Hard-Coded Musical Structure Object

[makevoice] contains the **gemnotes_counter** object - an external written in C that handles the creation and linking together of notation elements. With a system as complex as this it became clear that a single object was needed to handle the beamgroups, bars and the creation index of each object. The creation of beamed groups of notes on the fly, linking all objects to a staff abstraction, and ensuring that elements are horizontally spaced correctly are handled by this object. It is essentially a set of nested counters, and is presently capable of handling polyrhythmic note durations and user-defined groupings of notes within a bar, the automatic placement of barlines, tempo marks and time signatures.

Since this object handles the placing and index of each notation object, the introduction of new notation objects requires a certain amount of rewriting of the code, although graphical objects can be added to the GEM window outside of the [makevoice] structure. Imminent additions include dynamic marks, and articulation.

SCORE LANGUAGE AND TRANSLATION

Minimal Text Score Notation

The Gemnotes system translates a text-based score language (figure 3) into rendered notation elements on a staff. One of the design goals of the Gemnotes system was to keep the text-based score language as simple as possible. There are a number of commands that set the bar (and time signature if different from the last) note and tempo elements. Dynamics are the next feature to be implemented graphically, but the score language already contains MIDI velocities that may be interpreted as dynamics and articulation elements. The second reason for creating a minimal score language is that it make the process of real-time linking with pitch detectors etc, or translation of MIDI files much easier. To this end, a further set of objects has been created – **polyquant** that quantizes inter-onset times and durations with optional polyrhythmic complexity, and **gemnotes_barcount** which generates gemnotes scores from MIDI input. Note commands are formatted as a list: [“note” position duration pitch flat/sharp(0/1) velocity(. Work is underway to link together these two processes, so that music notation can be rendered directly from live input.

```
tempo 175 16 2 9 11 7;
bar 5 4 4 16;
note 0 58 2 0 40;
note + 70 2 0 46;
note + 72 2 0 51;
note + 65 2 0 58;
note + 74 2 0 63;
note + 64 2 0 70;
note . 76 2 0 70;
rest 8 0.5;
bar 4 4 4 16;
note 0 80 2 0 77;
note + 68 2 0 81;
note + 65 2 0 85;
note . 76 2 0 85;
rest 4 0.5;
note + 64 4 0 100;
end;
```

Figure 3: A section of a Gemnotes score. The bar commands also set the group lengths and note values for beam groups. A “+” means “next note” whereas a “.” means it is a chord with the previous note.

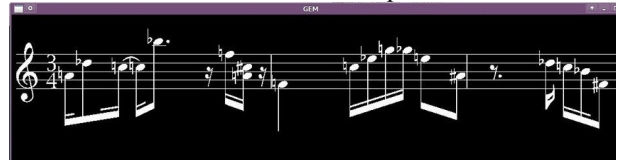


Figure 4: A complete line of notation.

CONCLUSION: NOTES ABOUT A SCORE

This system provides a basis for live score rendering using PD and Gem. A benefit of the system in question is that the material presented to the performer as a score can be re-interpreted in the electronic processing of the performer's audio, and may be reorganised indeterminately during performance. Scores may be created with a number of outcomes, or material may be arranged in a non-sequential manner using multiple staves for process music, or for stimulating improvisation, and the same pitches and rhythms used to set effect or playback parameters for a concurrent audio patch. Future aims are to integrate more notation elements, and to make the system less CPU-intensive. Also, live re-interpretation of material should be possible by directly rendering the output from a [fiddle~]-based pitch tracker for monophonic instruments.

REFERENCES

- Kelly, E. *Time in Music, Strategies for Engagement*, PhD Thesis, 2005, University of East Anglia, Norwich, UK.
- SIL International. *Open Font License (OFL)*, <http://www.sil.org/> 2005
- Allgeyer, R. *MusiQwik*, <http://cg.scs.carleton.ca/~luc/allgeyer/allgeyer.html>, 2007