UNIVERSITY OF WOLVERHAMPTON

Choosers

The design and evaluation of a visual algorithmic music composition language for non-programmers

Matt Bellingham Simon Holland Paul Mulholland

PPIG 2018

University of Wolverhampton

Algorithmic music composition

- The partial or total automation of music composition by formal, computational means (**???**);
- Structural elements such as indeterminism, parallelism, choice, multi-choice, recursion, weighting, and looping (**???**);
- Many musicians are not programmers and find existing tools difficult to use (**???**).

Problem setting — from PPIG 2014

- Existing tools require an understanding of programming languages;
- Many require an understanding of musical notation and/or music production equipment;
- Several programs impose working practices unconducive to compositional processes;
- In some cases the user was unable to define, and subsequently change, the musical structure;
- Complex graphical patches were hard to read and edit — spaghetti.

Design principles

New programming abstraction (the Chooser) to enable algorithmic music composition by non-programmers.

- Parsimony — a small number of consistent powerful ideas do the work combinatorially;
- Musically meaningful structuring actions are simple and quick to do;
- Both bottom-up and top-down construction are allowed in any combination;
- Affordances are designed for a wide range of users — children to experts — via progressive disclosure.

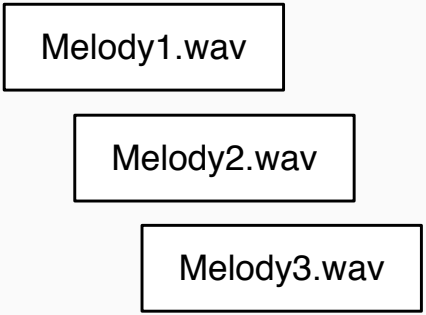


Figure 1: Samples are dragged in and shown as boxes.

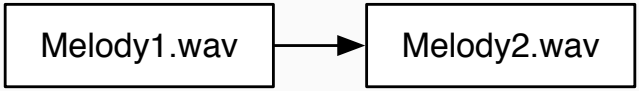


Figure 2: Sequence via arrows.

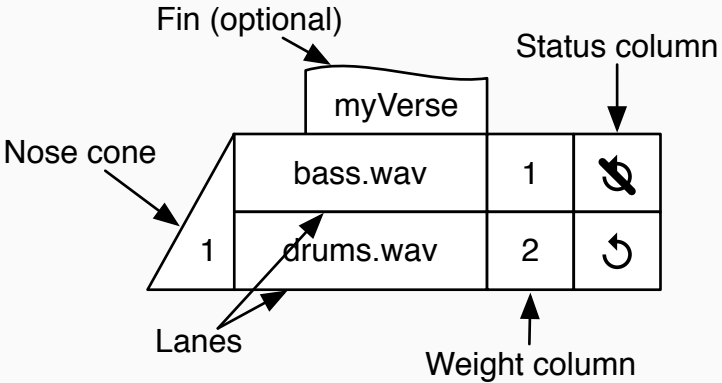


Figure 3: A Chooser which contains only soundable content is called a **Soundable Chooser**.

Example 1 — using phrases from *In C*

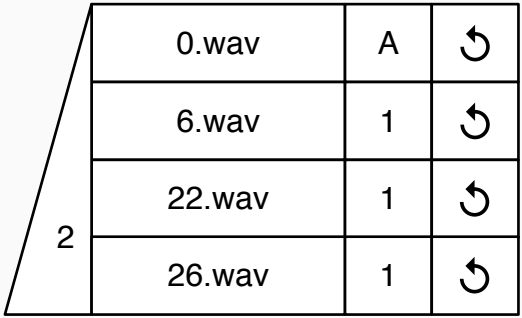


Figure 4: Two samples will be selected. `0.wav` is set to always be selected — therefore, one other sample will be selected and both will play concurrently.

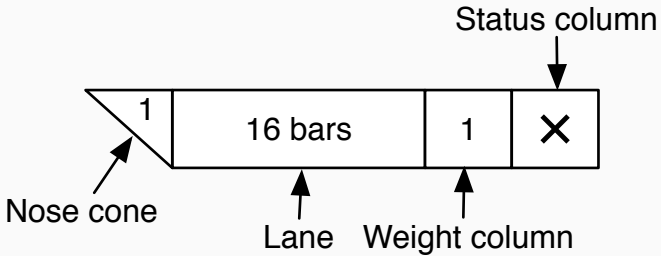


Figure 5: Time Choosers can be used to control the duration of a Soundable Chooser.

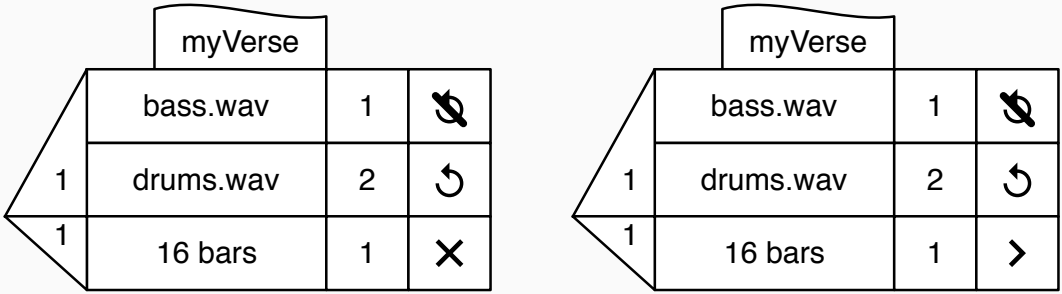


Figure 6: If a Time Chooser is attached to the bottom of a Soundable Chooser this produces a **Full Chooser**. Left — hard stop; right — soft stop.

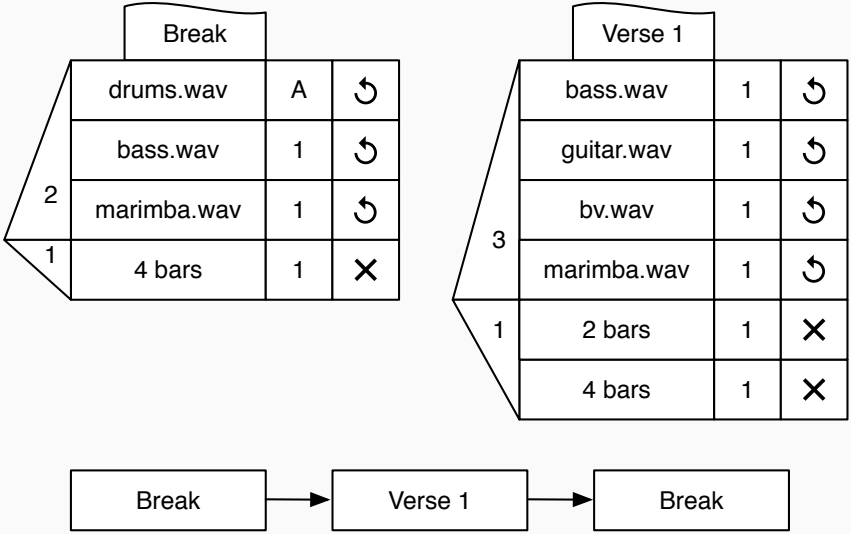
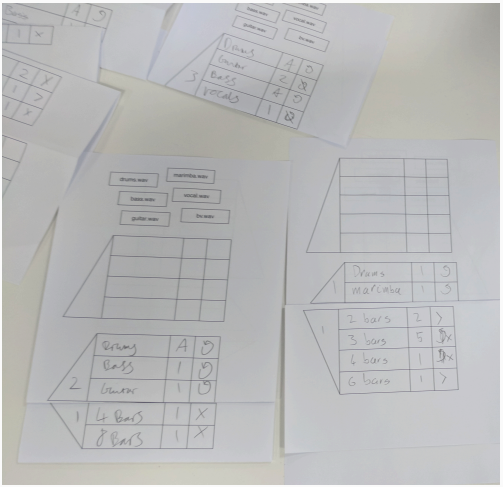


Figure 7: An example using samples by *Nine Inch Nails*. Note the multiple time lanes in the `Verse 1`

User tests

To test the ability of self-taught music producers to use Choosers to carry out a range of rudimentary algorithmic composition tasks; identify usability and user experience problems, tensions, and trade-offs.

- Seven pairs of users — participants were neither programmers nor traditional musicians;
- Users were active participants using the programming walkthrough method (**???**; **???**), including categorisation of issues into:
  - Questions;
  - Suggestions;
  - Observations.
- Wizard of Oz prototyping.



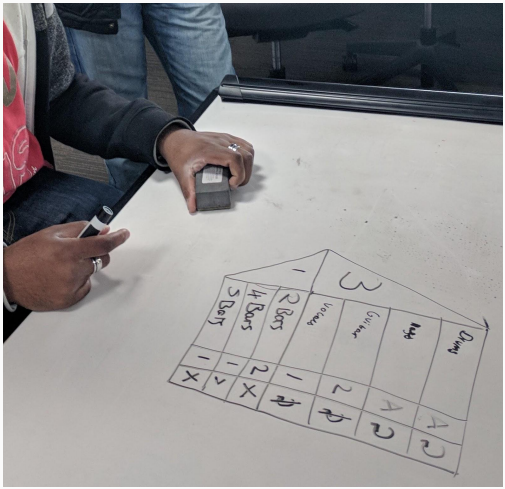


Figure 8: Users were given a range of practical tasks (reproduced in the paper) to complete on paper on on a whiteboard, with outputs played by the facilitator using a set of SuperCollider classes (???).

Results — musical issues

- All participants understood the system and were successful, with varying levels of assistance;
- Discussions on the desirability of algorithmic music;
- Stops and rests were initially confusing to some;
- Progress bar request — reasonable, but difficult in a nondeterministic system.

Results — programming-related issues

- Nose cone shapes were effective in communicating their combinatorial usage;
- Reuse or re-contextualisation of logic was observed, but didn't make sense in some contexts — the rationale behind these requests is instructive;
- Users required access to metadata.

Results — shared and existing knowledge

- A desire to leverage existing DAW knowledge and skill, which brought some frustration — expert in one environment, novice in another;
- Technological framing (**???**) and the expectations set by commercial DAWs as an influence on user requests.

Results — metaphor

- Useful when users are familiar with the original interface;
- Is it time to revisit some design assumptions in music production software?
- Hard and soft stops — no clear existing metaphor for a soft stop — both for the function and the icon.

Results — arithmetic

- Numbers are universally familiar and can concisely represent many relationships. Their use was motivated by parsimony and consistency. However …
- The use of numbers for multiple parameters was perceived as negative by three participants (‘too many numbers man!’).

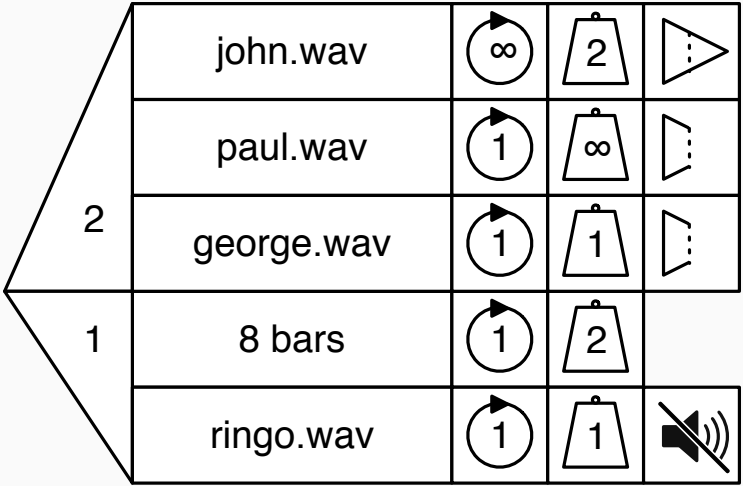


Figure 9: Each lane has a set number of repeats; ∞ can be used for repeats, weight, and nose cone; icons for repeats and weight; updated hard/soft stop icons; and playback/mute available for soundable content in the Time Chooser.

References i