

# Mondrian Music Description Language and Sequencer

Peter Brinkmann

brinkman@sci.cuny.cuny.edu

## Goals

- Create a flexible approach to musical notation that lines up notes sequentially, stacks lines of notes on top of each other, aligns such stacks sequentially, etc. In other words, structure music like one of Piet Mondrian's paintings.
- Create a keystroke-efficient macro language for music, inspired by  $\LaTeX$ .
- Turn text editors like Vim and Emacs into musical instruments.

## Front-end and back-ends

Mondrian consists of a *front-end*, the interpreter, and several *back-ends*, including a MIDI file writer as well as interactive sequencers that act as arpeggiators, pattern sequencers, etc. The sequencers can be used as editor plugins for both Vim and Emacs, allowing for instant execution of newly typed code.

## State machine

The interpreter has a *state* comprising user-accessible properties (current note, velocity, channel, track, etc.) and implicit properties such as time. The state also includes user-defined scales and macros. State changes within a pair of parentheses ( ) only affect the code inside the parentheses.

## Triggering note events

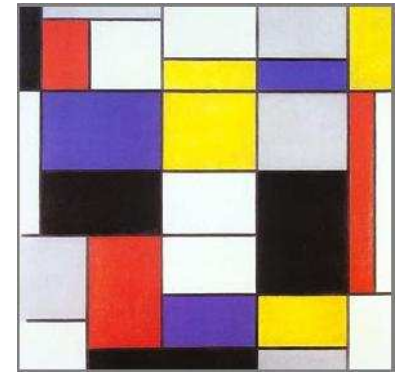
A number  $n$  that is not followed by a state-changing command triggers a note event (of the current pitch, velocity, etc.). If  $n$  has just one digit, then the duration is  $2^{-n}$ . For instance, 0 means a whole note, 1 a half note, etc. If  $n$  consists of more than one digit, then the last digit serves as an exponent, and the rest as a multiplier. For instance, 174 means  $17 \cdot 2^{-4} = \frac{17}{16}$ .

Frère Jacques

Dor - mez vous

Son-nez les ma-ti - nes

Ding ding dong



Piet Mondrian, Composition A

## Example: Frère Jacques

```
# define some macros
!(2 > 2 > 2 2< 2)!fj          # first line
!(2> 2 > 2 > 1)!dv           # second line
!(4> 3 > 3 < 3 < 3 < 2 2< 2)!slm # etc.
!(2 3< 2 3> 1)!ddd

# now play it!
2fj                          # one voice
[2dv | 2fj]                   # two voices
[2slm | 2dv | 2fj]            # etc.
[2ddd | 2slm | 2dv | 2fj]
```

## Some commands

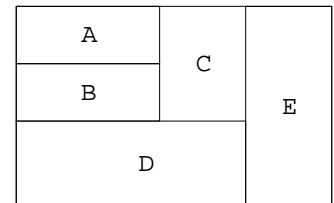
Mimicking the syntax of Vim, commands that don't trigger note events take the form  $nx$ , e.g., 5<, 3c, 120T.

>	move up $n$ steps in current scale
<	move down $n$ steps in current scale
v	set velocity to $n$
T	set tempo to $n$ bpm

Macro definitions are of the form `!content!label.`

## Lines and stacks

A note event is scheduled at the current time and advances the time by its duration, so that notes are played sequentially. Brackets [ ] and vertical bars | are used to stack such lines on top of each other. When Mondrian encounters an opening bracket, it saves the current time  $t_0$ . When it encounters a vertical bar, it resets the time to  $t_0$ . When it encounters a closing bracket, it advances the time to  $t_0$  plus the duration of the longest line enclosed by the brackets.



Lines and stacks:  
[[[A|B|C]|D]E