Live coding, live notation, live performance

Electronic Visualisation and the Arts, London 2016

Richard Hoadley
Digital Performance Laboratory, Anglia Ruskin University, Cambridge UK

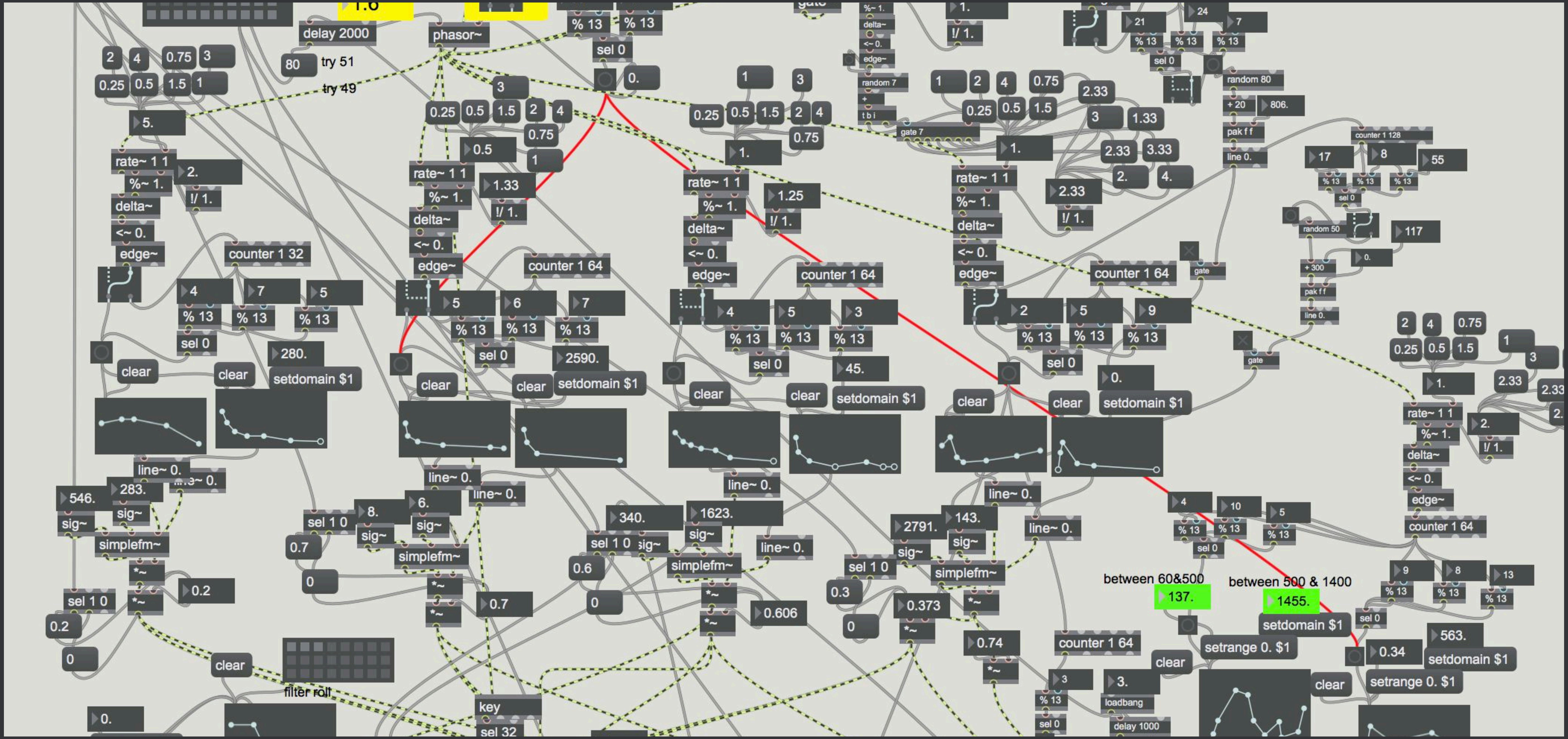
This presentation is available here: **http://rhoadley.net/presentations/eva2016.pdf**
This research has been subsidised by Anglia Ruskin University and Arts Council England
v0.04

- Now: talk and small demo
- Later: Demo/performance at 16:00 today in Wilkes 4 with Ian Mitchell, clarinet
- focusing on a variety of notations, but primarily music

- The written paper contains many more details than this talk
- The two poles defining this work are *authenticity* and *precision*, e.g.
    - baroque ornamentation.
    - a rock 'cover' version,
    - a Liszt 'edition' of Bach

History

- I'm an instrumental composer, working with dots but with an interest in electronic music
- more algorithmic than electroacoustic
- more live than fixed media
- interested in working alongside other expressive domains (dance, text, etc.)
- moved from 'standard' software towards programming, via MaxMSP:



https://cycling74.com/forums/topic/great-features-in-maxmsp-7-what-a-pity-about-the-interface4/

[5] http://rhoadley.net/presentations/eva2016.pdf

- and then on to SuperCollider (and to a lesser extent other environments).
- SuperCollider is a fully coded environment with an optionally available GUI allowing the process of programming to become a fully-fledged part of the composition and interaction process:

```
(
~lineFunc = ({ arg lineNum=80, notes=[[50, 65], [80, 90]], vels=[10, 60];
var scaleSpec = ControlSpec(80, 1, \lin, 0.01, 0.25);
var numScale;

numScale = scaleSpec.unmap(lineNum);

~edge[0].scale("edgeWin", "score", 1, numScale);

~edge[0].colour("edgeWin", "score", 1, [0,0,0,255]);
~edge[0].origin("edgeWin", "score", 1, -1, 0);
~edge[0].x("edgeWin", "score", 1, -1.5); // how far left to start the score
~edge[0].y("edgeWin", "score", 1, 0); // how far up to start the score

~multiPattDigiTask.stop; ~multiPattDigiTask.reset;
~multiPattDigiLines.value("~edge[0]", "edgeWin", 1, 1, false, lineNum, false, loopWait:0.03,
loopWaitRand:false, pitchInc: [1], pitchIncRand: false, loNote: rrand(notes[0][0], notes[0][1]),
hiNote:rrand(notes[1][0], notes[1][1]), loVel:vels[0], hiVel:vels[1], displayWhen:1, delay: 0.0, display: true,
play: true, itemNum: 1);
});
);

~lineFunc.value(14, [[10, 20], [80,90]], [40, 50]);
```

- These factors together allow for a particular, exploratory and experimental view of the links between programming, composition, performance. notation and improvisation.
- Some years ago now I met Nick Collins when he was doing his PhD in Cambridge – he was demonstrating the SC BBCut library as well as a thing called 'live coding' as well.
- Live coding is an interesting activity not least because it investigates certain music and code-based practices in a new way.
- These practices include programming, composition, performance. notation and improvisation.

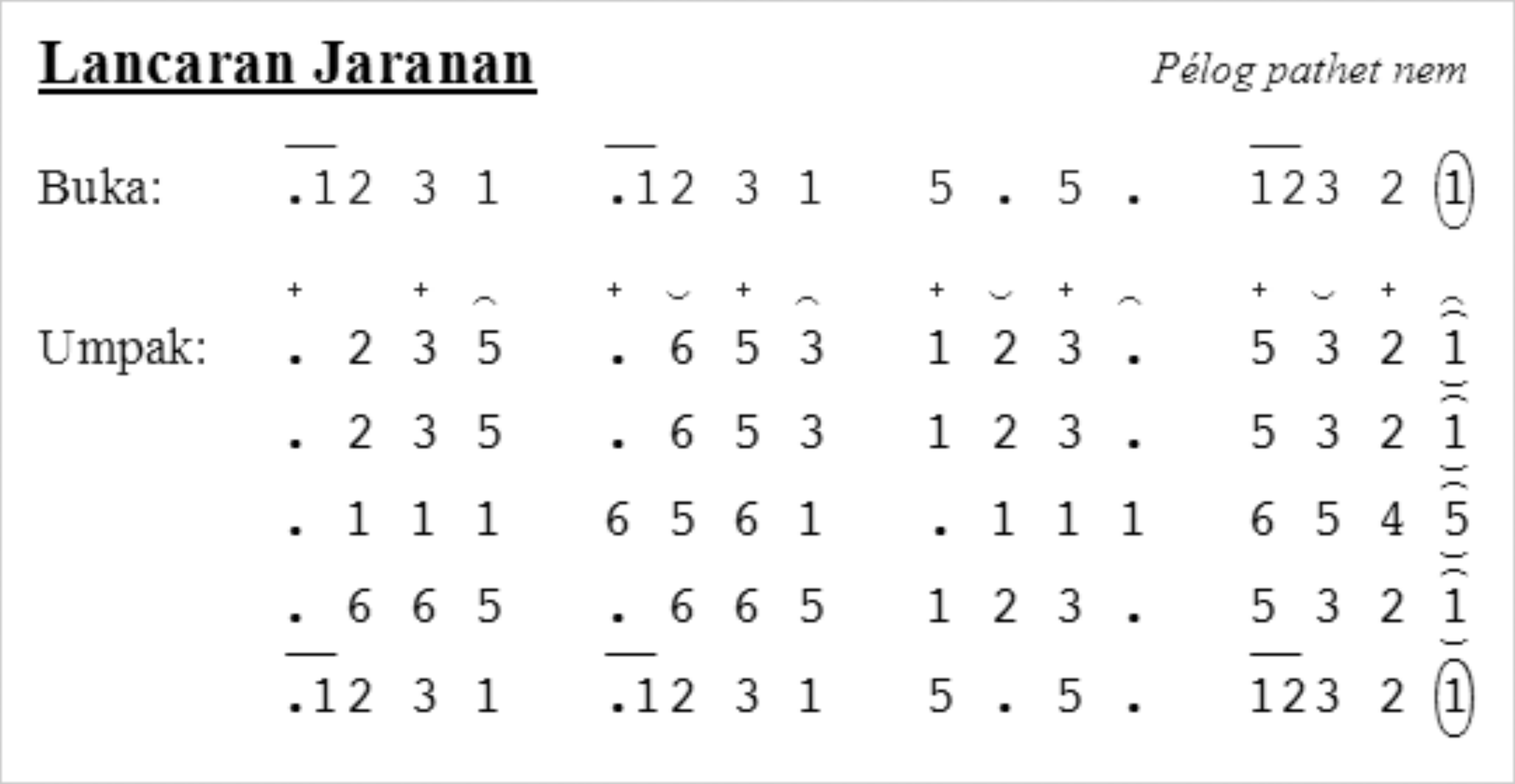
- However, it also exposes some issues which equally interest me and which I more fully investigate in the written version of my paper.
- These include the human-computer interface (qwerty keyboard, anyone?) and the nature of interaction with software itself.

Artistic statement:

- For what it's worth, I think that computers are not, and can never be, the same as 'acoustic' musical instruments, and that code can never be the same as musical notation (of any sort). However, computers and the software they run can and will provide a fascinating, expressive and possibly dominant role over time.

- I, in the meantime, am interested in the intersection of code and music notation itself.
- While I'm currently concentrating on western common practice notation, I fully intend to explore other notations, such as experimental western notation, other music notations from around the world and other notations in general, such as language and dance.
- Here are some quick examples:

Music and Notation: Gamelan



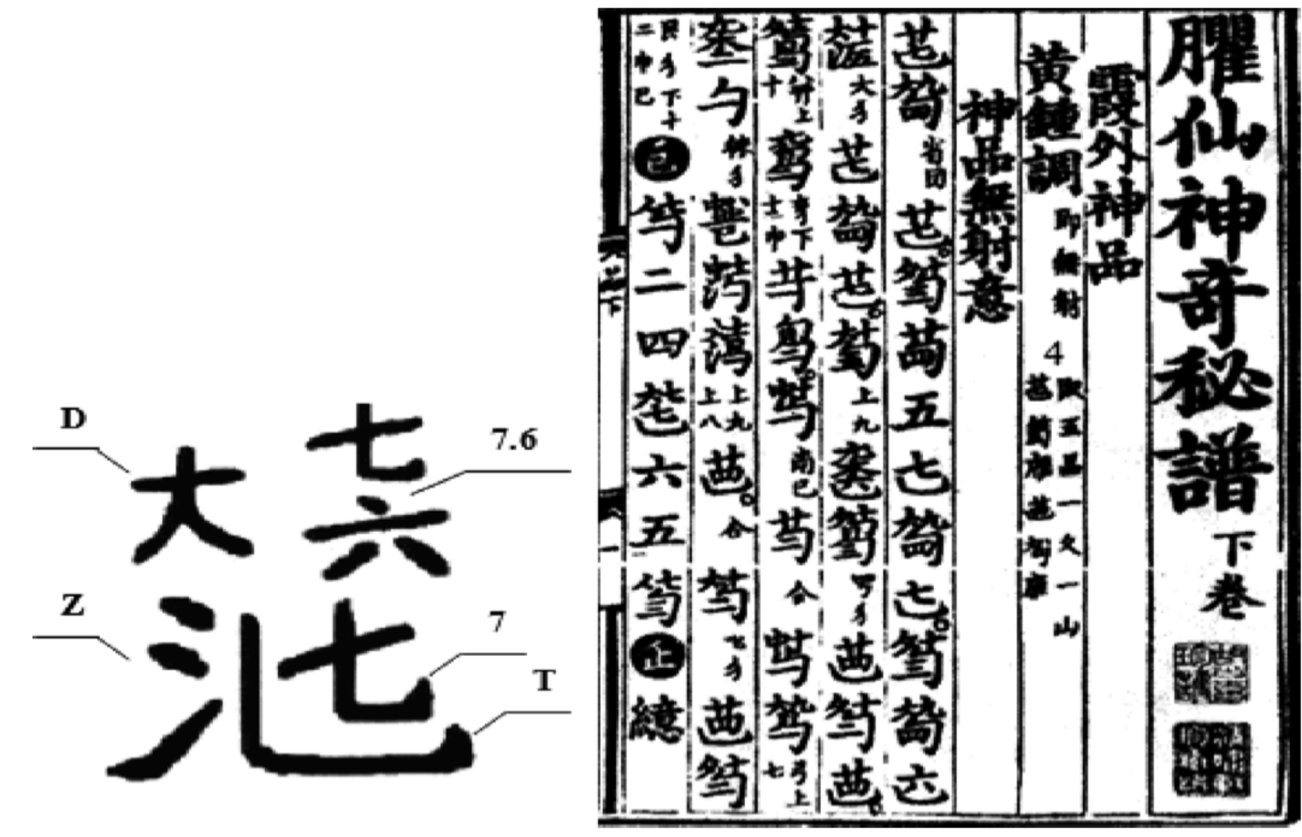
Lancaran Jaranan *Pélog pathet nem*

```
Buka:    .12 3 1    .12 3 1    5 . 5 .    123 2 1

         +   +   ⌢  +  ⌣  +  ⌢  +  ⌣  +  ⌢  +  ⌣  +  ⌢
Umpak:   . 2 3 5    . 6 5 3    1 2 3 .    5 3 2 1
         . 2 3 5    . 6 5 3    1 2 3 .    5 3 2 1
         . 1 1 1    6 5 6 1    . 1 1 1    6 5 4 5
         . 6 6 5    . 6 6 5    1 2 3 .    5 3 2 1
         .12 3 1    .12 3 1    5 . 5 .    123 2 1
```

http://rhoadley.net/presentations/eva2016.pdf

Music and Notation: Guqin tablature

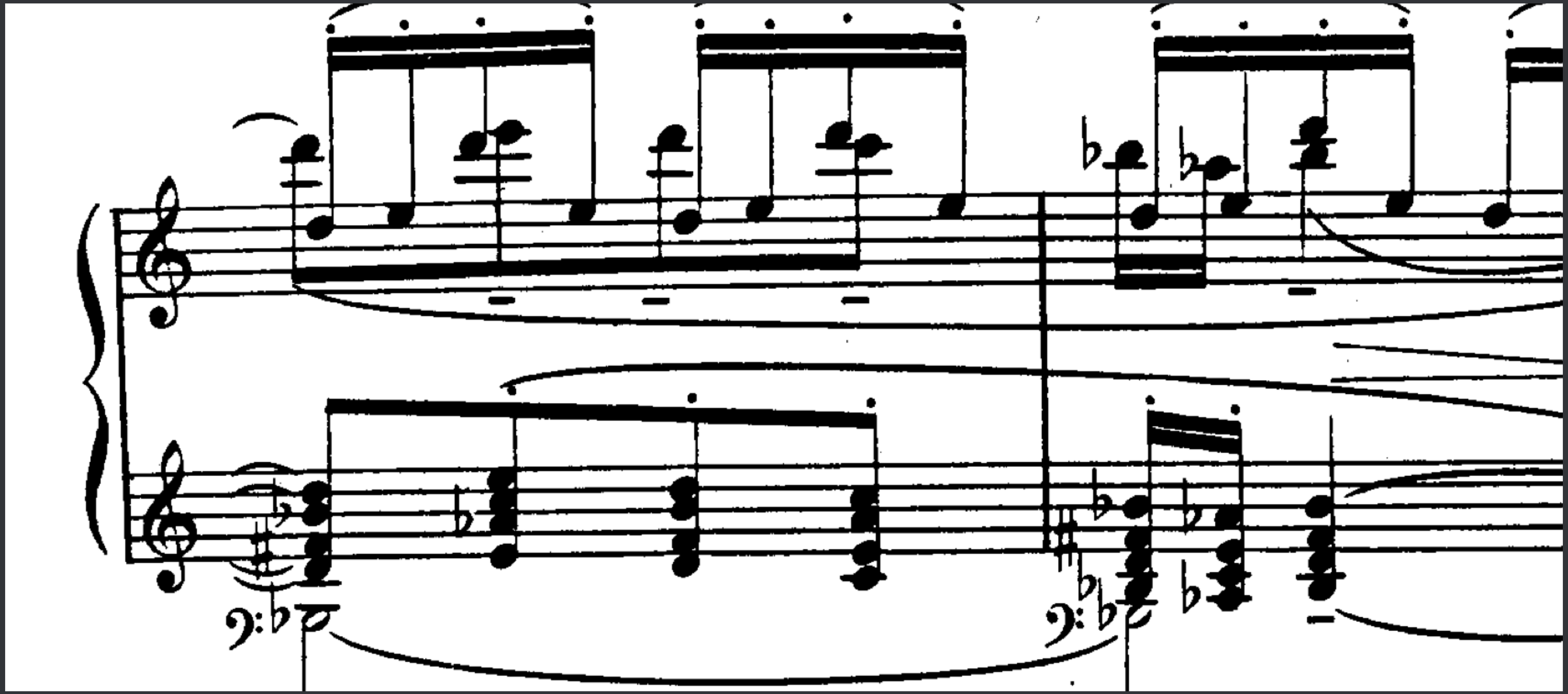


The “D” part denotes the finger used to press the string, and the “7.6” part denotes the pressing position. The “Z” part denotes the sound decoration method, the “T” part denotes the plucking method, and the “7” part denotes the string being played. The overall meaning of this symbol is that the player uses left thumb to press on the 7.6 position of the 7th string, uses the right forefinger to pluck the same string outward, and kneads the string while pressing. Notice that other symbols may contain [a different number of] parts. The right side shows a piece of tablature that consists of many different symbols. (Sun et al., 2010)

Music and Notation

"Traditional notation is biased towards music that is humanly performable. This … is an obstacle when trying to notate music intended for computer performance, where the notation is often found to be deficient, inconsistent, and redundant." (Hudak et al. 1993)

Music and Notation: Impossible



Debussy *Violes* (date): impossible notation (figure 3)

Oramics

As an aside, this is an interesting recent example of notation, programming and performance (but crucially no 'interpretation') being mixed up:

http://www.bbc.co.uk/news/technology-36651270

Live Coding

- There are a variety of definitions of live coding in a variety of places (e.g. TopLap)
- "Live coding is making changes to algorithms as they run, with the possibility for both live feedback and a live audience. (See http://iclc.livecodenetwork. org/2016/)"
- "changing the rules as part of performance"

Although there's often an emphasis placed on improvisation in live coding...

"Improvisation is not quite as free as it might appear because there are rules. I generally make techno and that has quite fixed rules about what you can and can’t do ... In terms of the algorithm I have lots of code that I use and don’t change during a performance but I suppose I could if I wanted to I might have an idea for something and try it out. It’s Turing complete so in theory anything can happen but in practice it’s probably quite a limited exploration. (McLean 2013, 35:05)"

https://vimeo.com/69687342

Live coding

- A key aspect of live coding has been 'show us your screens':

https://vimeo.com/20241649
(from around 20 secs). Dan Stowell at 2:00

Show us your screens 3:20, 5:00
'expose the process'
' people get a bit of a feel for it'

Show us your screens

‘Obscurantism is dangerous. Show us your screens.’ http: //toplap.org/wiki/ManifestoDraft. The Toplap website also announces that “All code manipulation is projected for your pleasure.” (McLean et al. 2004), also, presumably to discourage any nefarious use of functions or software.

BUT you have to be able to show a certain amount at once… If you're using lots of code how can this work?

Show us your screens

- But, understanding code quickly restricts the complexity of the code
- The code structure can undermine ease of understanding or presentation
- How does the process *really* compare to an understanding of music (Dan's description of seeing a guitarist).
- The use of notation itself plays a role in the type of music created. Purely improvised music can reach a certain level of a certain type of complexity and detail, meaning that never-notated (but not improvised) music tends to reach a certain level of complexity.
- Sets of libraries...

Types of music based on notation/non-notation:

- Originally notated music (e.g. Gamelan, Bach, Guqin, eventually having legal force from 19th century)
- Notated in order to enable synchronisation
- Notated in order to explore the possibilities of notation
- Orally/aurally transmitted but primarily unimprovised music (e.g. pop, rock, some jazz)
- Orally/aurally transmitted but primarily improvised music (e.g. some jazz)
- Transcribed music (originally unnotated, e.g. ABBA 'scores')
- Improvised (and mainly not transmitted) music ('free' improvisation)
- These distinctions also apply in different ways to other notations, such as text and dance.

The tools

- provide a structure for the **generation** of music and/or common practice **notation** as well as text and **graphical** elements (including raster and vector images).
- facilitate communication between **SuperCollider** and **INScore**
- offer the beginnings of a more standard interface for **physical mapping** and **live notation**

which are located…

- https://github.com/supercollider/supercollider
- http://inscore.sourceforge.net/
- http://rhoadley.net/inscore (on request)

Some examples

Edge Violations

Projection as score

https://vimeo.com/169107960

(~9:30)

Forthcoming events

- Edge Violations with Ian Mitchell, clarinet, Electronic Visualisation and the Arts, Covent Garden, London, July 12th 2016
- Paper: Live Performance with Live Scores, Performance Studies Network Conference, Bath Spa University, July 15th 2016
- Paper: Intersemiotic translations and live notations in dance, music, poetry and graphics, Music and Visual Cultures Conference, Maynooth University, Ireland, 21st-23rd July 2016
- Performance: Calder's Flute, Sound and Music Computing, Hamburg, 3rd September 2016
- How To Play the Piano (Paper)
  ICMC 2016, HKU University of the Arts Utrecht, 12th-16th September 2016

Thank you

any questions?

contact:
research@rhoadley.net

this presentation is available at
http://rhoadley.net/presentations
as **eva2016.pdf**

video recordings of past performances are at **rhoadley.net/youtube** and **rhoadley.net/vimeo**