

Live Coding: an Overview



IOhannes m zmölnig, Gerhard Eckel

zmoelnig@iem.at

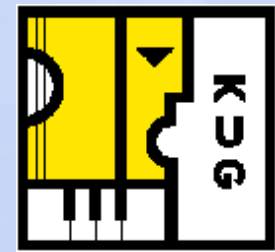
Institute of Electronic Music and Acoustics

University of Music and Dramatic Arts

Graz, Austria



- University of Music and Dramatic Arts, Graz
- <http://www.kug.ac.at>





Overview

- Introduction
- Prerequisites
- History
- Issues and Quests
- Conclusions

Synonyms



- Live Coding
- Just-In-Time Decisions
 - JITlib/SuperCollider
- On-the-Fly Programming
 - Chuck



What is Live Coding?

- writing „code“ before an audience
- performance practice
 - reaction to classical electronic music
 - tape music
 - laptop music (tic tac toe!)

Introduction



Software Art / Code Art

- Focus on Algorithms
 - ~~what is created?~~
 - **how** is it created?

Introduction



- improvisation
 - improvising structures instead of phrases,...
- expertise
 - writing algorithms
 - using environments as instruments

Prerequisites



- bound to general purpose computers
 - powerful
 - alternatives
 - live coding without computers
 - algorithmic contests
- bound to audience
 - „small“ computers
 - visible code

History



- 1970s (Microcomputers)
- 1985
 - Ron Kuivila (STEIM)
 - 1st Live Coding Session
- 1980s
 - The HUB
 - showing code

History



1990s

- Internet
- Open Source

- net.art / Software Art / Code Art
- laptop music

- computers fast enough
 - re-build DSP graphs without dropouts
 - use interpreters for audio generation

History



2000

- SLUB (A. Ward, A. McLean)
- Parcel/JITlib (Julian Rohrer)

2000-

- dedicated environments/languages

2004

- TOPLAP

Quests



- Expressivity
 - individual
- Code Literacy
 - public
- Collaboration
 - collective

Quest for Expressivity



- „Live Coding is about *algorithms* rather than *tools*“
- Code as Medium
 - primary artistic outcome: algorithms
- use „elegant“ languages:
 - abstraction, recursion,...
 - functional programming

Quest for Expressivity



- genius
 - „master of algorithms“
- anti-aesthetics
 - geek culture
 - „slow code“

Quest for Code Literacy



„Show us your screens“

- code projection
- non-traditional performances (laptops within audience)
- *audio* is better suited than video
 - no interference with „visual“ code

Quest for Code Literacy



- reading algorithms = reading code
- audience
 - usually code-illiterate!
- reading code per se difficult
 - On-the-fly programming even more so!
 - few comments
 - condensed solutions harder to grasp
 - moving focus

Quest for Code Literacy (cont.)



-
- text
 - linear medium
 - (structured) code
 - non-linear
 - more-dimensional representation!

Quest for Code Literacy (cont.)



- graphical languages
 - Pd, Max/MSP,...
- simple metaphor
 - dataflow
- badly structured
 - dataflow vs. controlflow
 - control structures hard to read (even for experts) [Petre,1995]

Quest for Code Literacy (cont.)



- abstract representation
 - no direct visualization of coding
 - visualize *structures* within code
- example
 - ChuckK/audicle
- drawback
 - what happens when doing „visuals“?

Quest for Collaboration



- improvisation in groups
- sharing of algorithms/code
 - beyond the playing/hearing/reacting cycle
- examples:
 - „blind date“ (pd-graz)
 - collaborative editor
 - „powerbooks unplugged“
 - run code on remote machines

Conclusions



Live Coding

- a *new* performance practice
- using software-environments as instruments rather than ready-made applications
- improvisation on a structural level rather than on a musical level
- go see the performances at ICMC!

Live Coding: an Overview



Thanks for your attention!