Introduction to livecoding with Mercury

Timo Hoogland

General info

length: +/- 3.5 hours

participants: Max. 20 people (preferably < 15)

ages: 12+

skills: no programming or musical experience required

computer: Bring your own laptop and headpones

language: English (spoken & written, optional in Dutch)

system requirements:

- Mac:
  - OS X 10.11.6 or later
  - Intel Core i5 processor (or faster recommended)
  - 4 GB RAM (8 GB or more recommended).
- Windows:
  - Windows 7, Windows 8 or Windows 10,
  - 64-bit Intel or AMD multi-core processor (Intel Core i5 processor or faster recommended)
  - 4 GB RAM (8 GB or more recommended).

Workshop description

During this workshop you will be introduced to the livecoding environment Mercury. Mercury is a minimal and human-readable language focussing on quick expression in composition, performing and communication of livecoded music. Mercury is a great environment to get introduced to the amazing worlds of live coding, creative coding, algorithmic composition, electronic music and of course the Algorave scene! In this workshop you will learn how to generate different rhythms with algorithms and probabilities, play sound samples from the computer, how to make basslines with a synthesizer and program melodies for arpeggiators by combining various algorithmic composition methods.

The development of Mercury is made possible by funding from the Creative Industries Fund NL.

In-depth description of Mercury

In Mercury, all elements of the language are designed around making code more accessible and less obfuscating for the performer and the audience. This motivation stretches down to the coding style itself which uses clear descriptive names for functions and a clear syntax. Furthermore the editor is restricted to 30 lines of code, keeping all code always visible. Mercury provides the performer with an extensive library of algorithms to generate or transform numbersequences that can modulate parameters over time.

Topics

- What is livecoding?
- How is it used by artists?
- What is Mercury?
- What is Mercury for and how is it different or similar to other livecoding languages? What is the concept and vision behind the language?

Goals

Starter Goals

- Learn to play a single sample
- Change the speed of the played sound
- Add another sound
- Make a beat

Intermediate Goals

- Learn to play a single synth
- Make a melody for the sound
- Make a rhythm for the sound
- Use functions to generate melodies and rhythms
- Add a rhythm through different speeds or randomness

Advanced Goals

- Introduction to algorithmic composition
- Combine functions to generate more complex melodies and rhythms

- Add effects-processing for sounddesign
- Modulate parameters of effects

Schedule (by approximation)

30 minutes :

- Downloading the software
- Introduction to livecoding
- Short demonstration
- Installing / Trouble-shooting

30 minutes :

- Syntax Explanation
- Play a single sound
- Work on starter goals

30 minutes :

- Play and explore possibilities with rhythms

15 minutes :

- Break

30 minutes :

- Make a melody for a sound
- Work on intermediate goals
- Use ring transformation methods to extend and change melodies/rhythms

30 minutes :

- Play and explore!
- Individual help
- Optional move to advanced goals

30 minutes :

- Show & Tell by participants

Technical Requirements

Sound:

- Speakers and Mixer (PA) with stereo-DI or mini-jack

Visual:

- Projection and screen with HDMI connection

Participants:

- Tables
- Chairs
- Power outlets on table

Consumptions:

- Coffee / Tea / Water

Thanks

- Mercury has been granted funding from Creative Industries Fund NL
- Mercury is sponsored by Creative Coding Utrecht