The i-score interactive sequencer

an intermedia sequencer for interactive scenarios authoring

Jean-Michaël Celerier, Théo de la Hogue

LaBRI, Blue Yeti, GMEA

January 30, 2016

The problem

- A lot of tools for entirely fixed temporal content
  $\rightarrow$ traditional song-making.
- A lot of tools for fully interactive content
  $\rightarrow$ artistic installations.

- What goes in between ?

The problem

- A lot of tools for entirely fixed temporal content
  $\rightarrow$ traditional song-making.
- A lot of tools for fully interactive content
  $\rightarrow$ artistic installations.

- What goes in between ?

The problem

- A lot of tools for entirely fixed temporal content
  $\rightarrow$ traditional song-making.
- A lot of tools for fully interactive content
  $\rightarrow$ artistic installations.

- What goes in between ?

Futuroscope, France : the Sprinter

Credits : Blue Yeti

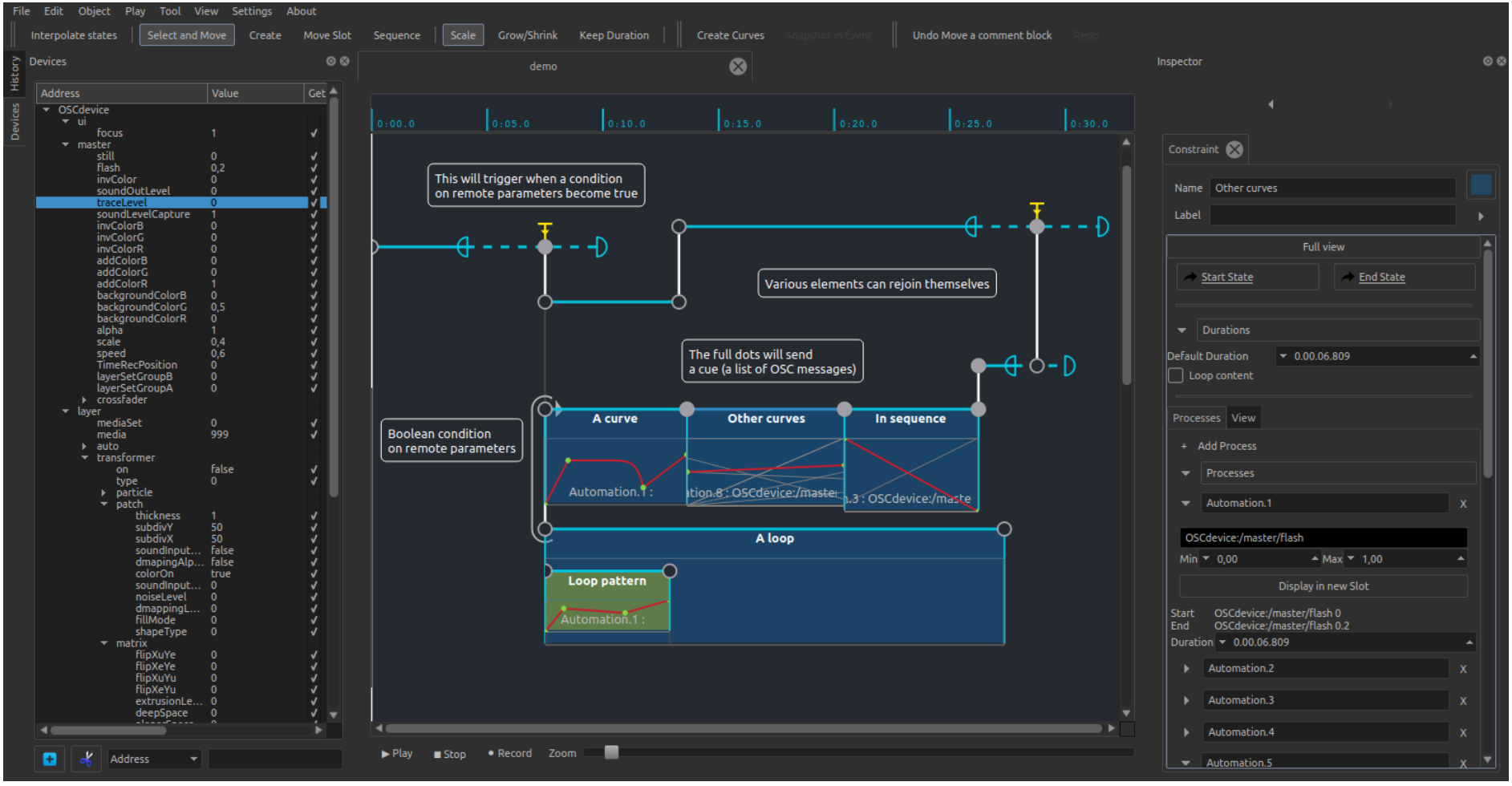
Tumbleweed

Credits : Les Baltazars

Jamoma

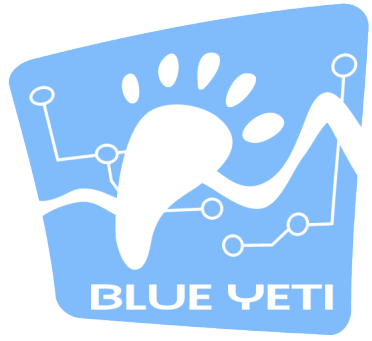

MODELS_patcher
input~.model in
equalizer~.model eq
output~.model out
VIEW_patcher
/in
soundfile
No file selected
0 : 0 : 0
Transpose
0 st
Open
Play
Loop
Pan
0.00
Master Gain
100.00
Audio Off
DSP
/eq
equalizer~.view.large /eq
/out
Saturation
Depth
10.0 %
Limiter
Preamp
6.0 dB
Release
300 ms
Record
aiff
int24
Audio Off
0.00
DSP
1
2
Pan
0.00
Master Gain
100.00
equalizer~
/eq
24
18
12
6
-6
-12
-18
-24
1
2
3
4
Type
peaknotch
Frequency
1.24 kHz
D#5
Gain
18 dB
Bandwidth
1.00 Octaves
View Phase
Add Band
Remove Band

The software



Contributors, Companies, Agencies involved

LaBRI
www.labri.fr



Blue Yeti
www.blueyeti.fr

GMEA
www.gmea.net

le cnam

CNAM :
CEDRIC, ENJMIN
cedric.cnam.fr

ISTS
ists-avignon.com

ENSATT
ÉCOLE NATIONALE SUPÉRIEURE DES ARTS ET TECHNIQUES DU THÉÂTRE

ENSATT
ensatt.fr

Artists: Les Baltazars, Renaud Rubiano, Antoine Villeret...

What i-score is :

- A visual **programming language**
  $\rightarrow$ Conditions, loops, structuring, in a timeline

- Free software : **GPL v3** (UI) & LGPL v2.1 (Engine)
- Built in **C++** (Qt, CMake)
- Available on Linux / OS X / Windows

- Alpha-quality ☹

What i-score is :

- A visual **programming language**
  $\rightarrow$ Conditions, loops, structuring, in a timeline

- Free software : **GPL v3** (UI) & LGPL v2.1 (Engine)
- Built in **C++** (Qt, CMake)
- Available on Linux / OS X / Windows

- Alpha-quality ☹

What i-score is :

- A visual **programming language**
  $\rightarrow$ Conditions, loops, structuring, in a timeline

- Free software : **GPL v3** (UI) & LGPL v2.1 (Engine)
- Built in **C++** (Qt, CMake)
- Available on Linux / OS X / Windows

- Alpha-quality ☹

What i-score is not :

- PureData (yet)
- Ableton Live (yet)
- Bug-free (yet ! ☺)

Does not operate on its own !

- It’s a control center

What i-score is not :

- PureData (yet)
- Ableton Live (yet)
- Bug-free (yet ! ☺)

Does not operate on its own !

- It’s a control center

What i-score is not :

- PureData (yet)
- Ableton Live (yet)
- Bug-free (yet ! ☺)

Does not operate on its own !

- It’s a control center

What i-score is not :

- PureData (yet)
- Ableton Live (yet)
- Bug-free (yet ! ☺)

Does not operate on its own !

- It’s a control center

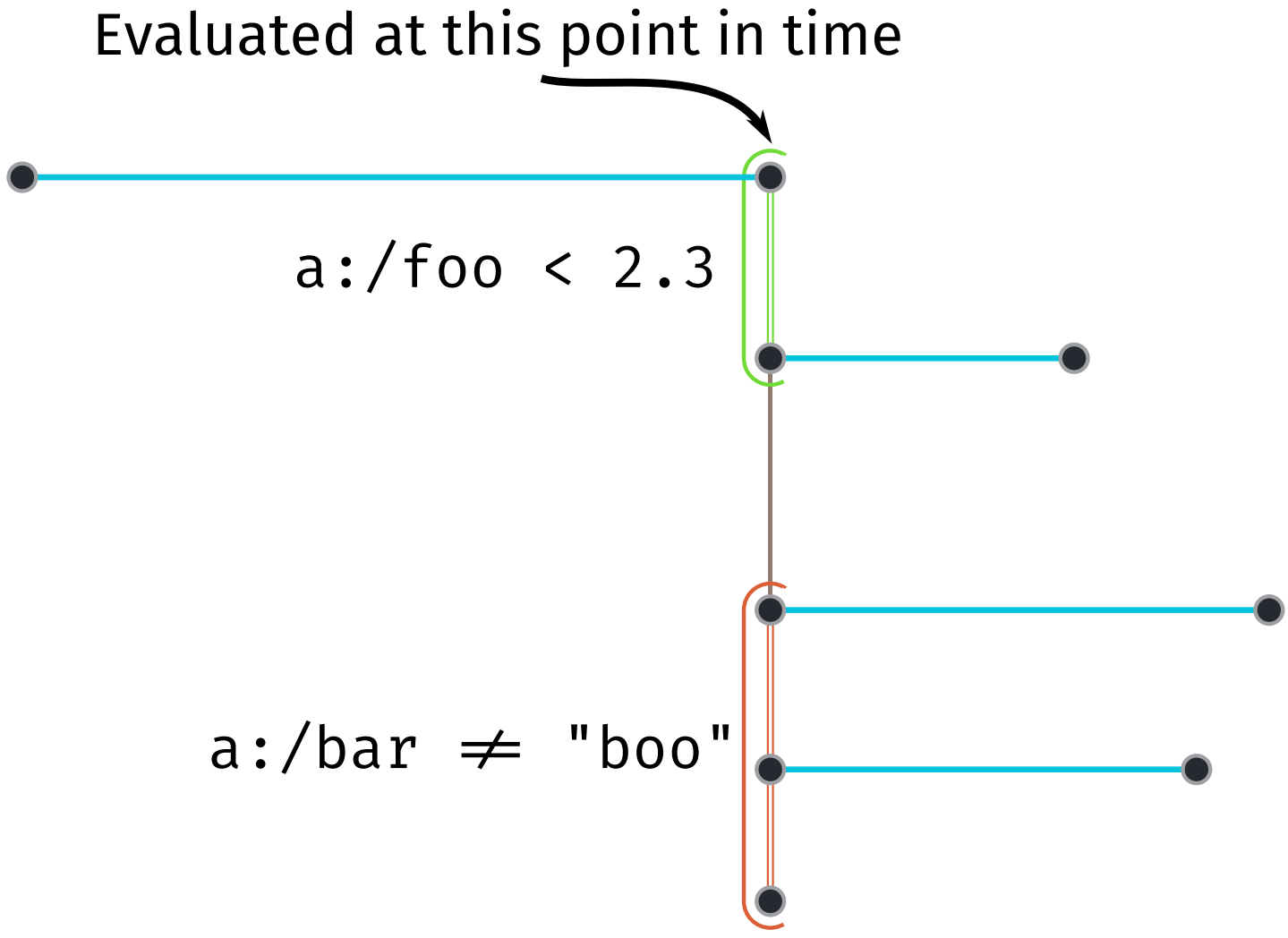
Demonstration

Inter-operability

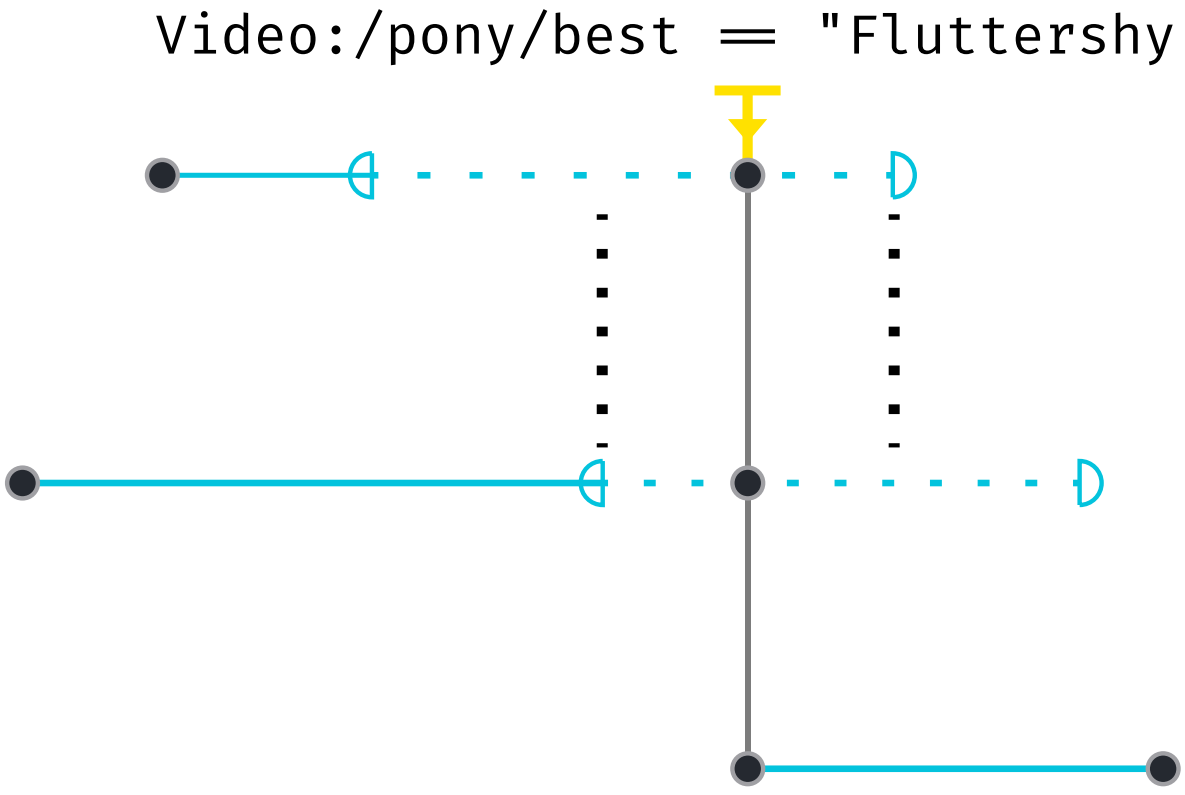
- Compatible environments :
  Max/MSP, **PureData**, Unity3D, **OpenFrameworks**, **Processing**, **Jamoma**, Modul8, Millumin, Quartz Composer, **Qt**...
- Anything that communicates over **OSC**.
- Extensibilty via **plug-ins***.

*API not stable until v 2.0

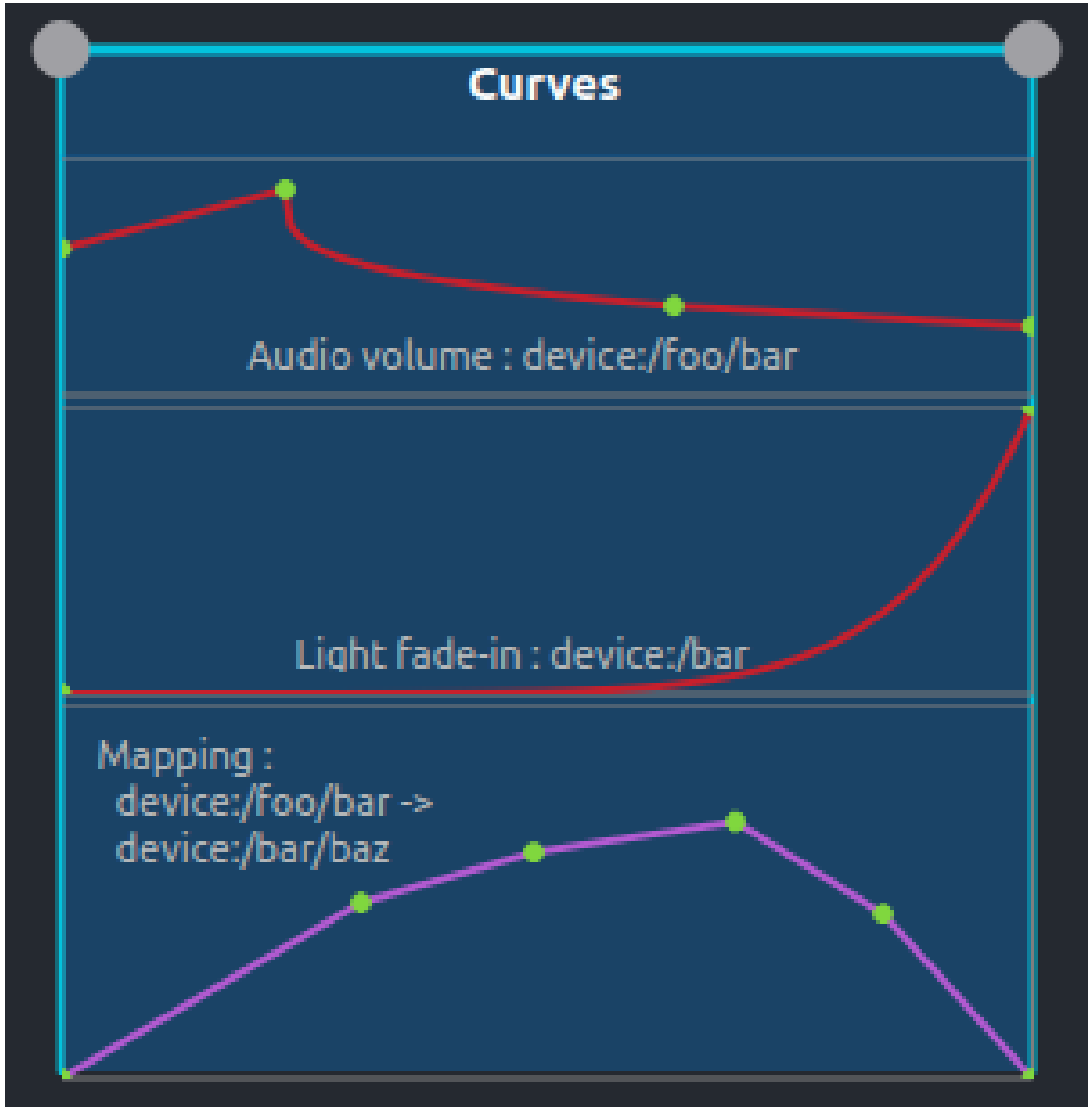
Conditions



Triggering



Automations, mappings



Various kinds of curves

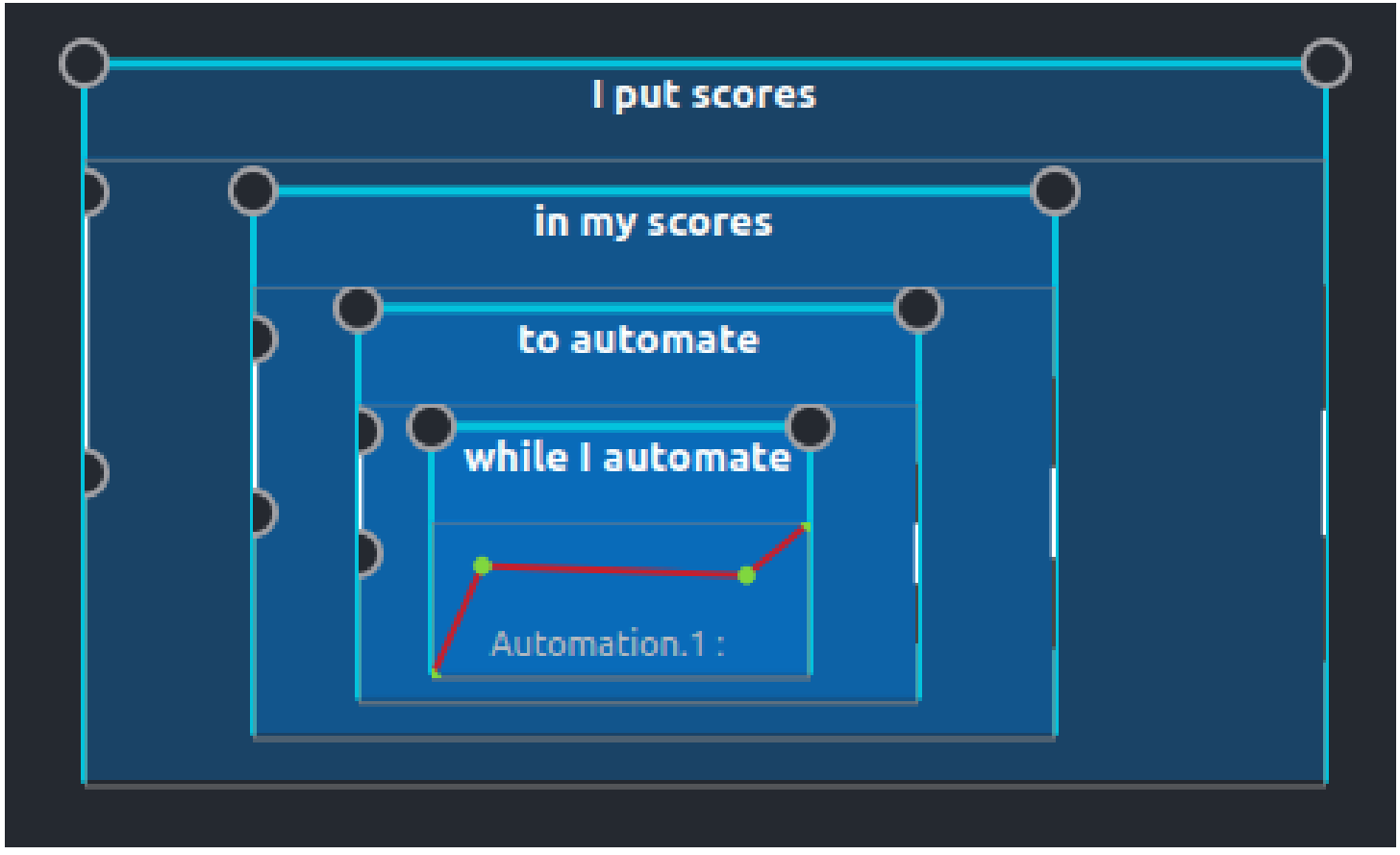
JavaScript

```
function(t) {
    var obj = new Object;
    obj["address"] = 'dev:/foo/bar';
    obj["value"] = t + iscore.value('other:/baz');
    return [ obj ];
}
```

Will get called at each tick

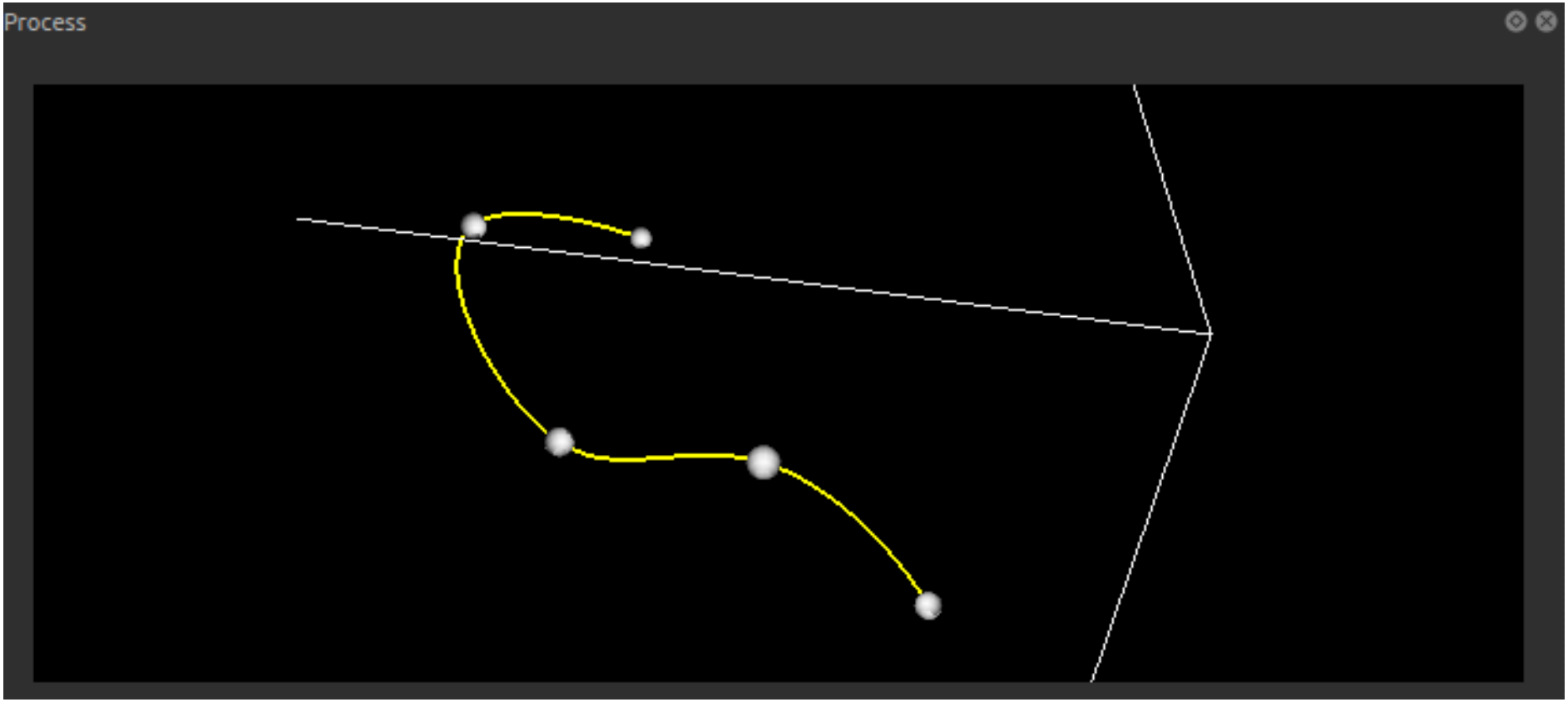
- Uses Qt's QJSEngine.
- For now API with a single function : fetch a remote value.

Hierarchy



Scenarios can be **nested** arbitrarily

WIP : Spatial automations



- **3d splines** that uses VTK. Can be used to create paths in space for instance.
- **Spatial mappings** to compute collisions, distances, etc. and performs actions according to the result of such computations.

Future : distribution ?

- Currently : multiple instances can work together at the editing stage.
- In progress : distributed execution.
- Example scenarios :
  - 100 phones controlling a parameter together.
  - Live backups if a computer dies during performance.
  - Offloading due to performance requirements.

Future : other features

- **MIDI**, **WebSockets** support
- Some level of **patching**, like Pd
- Complete **remote-control** abilities.
  Currently : execution can be followed via a web page.
- Port execution engine to **FPGA**.
- Audio engine ?

Contributing

- **UX**, **UI** (mock-ups were done but not entirely implemented)
- **Documentation**, writing demo scenarios
- **Translations**
- Implement the **Minuit** protocol in your software with the OSSIA API
- Many ”low-hanging fruit” TODOs
- Mobile devices ports :
  - **Android** : builds and run but requires adapted UI.
  - **Web port** : with PNaCl, runs but crashes. Will open the way to WebAssembly.
  - **iDevices** (many artists use them).

Links

- **Grab a release !**
  `github.com/OSSIA/i-score/releases`
- **Protocols and implementations** :
  `github.com/OSSIA`
- **Official website (not up-to-date)** :
  `i-score.org`

Thanks ! Questions ?

Credits: 'simple' Beamer theme, Facundo Muñoz; Fira font