

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

Example installations



Figure: Futuroscope, France. Blue Yeti

Example installations

robots ensatt pascal

Screenshot

Agencies involved

ANR

LaBRI

GMEA

Blue Yeti

What i-score is :

- ▶ Visual programming language for interactive multimedia applications
- ▶ Free software : GPL v3 (UI) & LGPL v2.1 (Engine)
- ▶ C++ (Qt, CMake)
- ▶ Linux / OS X / Windows
- ▶ Alpha-quality ☹

What i-score is not :

- ▶ Audio sequencer
- ▶ Video sequencer
- ▶ PureData or Ableton Live
- ▶ Bug-free

Inter-operability

Max, PureData, Unity, OpenFrameworks, Processing, Jamoma, Modul8,
Millumin, Quartz Composer, Qt...

Demonstration

Automations, mappings

Javascript

Hierarchy

Spatial automations

Future : audio ?

Future : distribution ?

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.

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

Thanks ! Questions ?

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