PushInterface.jl - Interacting Julia with Electronic Musical Instrument

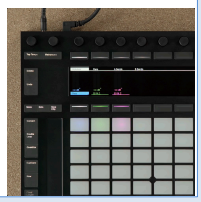
WooKyoung Noh

Abstract

Ableton Push is a popular electronic musical instrument. I have created a Julia interface to control this. Playing with the instrument, I ran the functions of the other Julia packages. The smooth and flexible operations among the hardware and Julia packages, that demonstrates the productivity and the real comfort in Julia programming.

Motivation

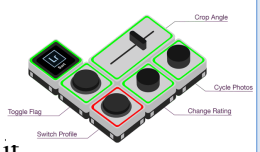
I thought that recent visualization tools in machine-learning area seemed to be very similar to manipulating the signals of music software.
In the meanwhile, I liked to be good at Beat Making (making beats in hip-hop and electronic music), so I dealt with electronic music and related equipments.
One day I repeated the experiment, I found that the knobs and pads on electronic instruments look like the sliders and buttons of GUI programs.
Keeping this in mind, I made a Julia interface for Push.
PushInterface.jl is the implementation of the Ableton Push 2 MIDI and display interface.
To that way, it could be applied to not only music but also being useful in general user interaction.



Background

MIDI (Musical Instrument Digital Interface) was standardized in the early 1980s.
And now, MIDI-based electronic musical instruments are indispensable in modern music production.
Of course, Ableton Push has well supports MIDI.
PushInterface.jl depends on rtmidi library to work with MIDI API.
Julia's ccall makes easy to call functions for C-exported library.

There's another interesting product Palette. It is a modular hardware device for fast editing like photo editing and lighting.
I don't have it yet, but I want to try it by all means.



Results and Discussion

After catching the MIDI events successfully, I could make some examples with various Julia packages.

* Interactive websocket communication with Bukdu.jl
Bukdu.jl is a web development framework based on HTTP.jl. MIDI events like changing knobs are delivered to the web page through the websocket. Additional DOM manipulation with JavaScript would be possible.

* Insertion data with Octo.jl
Octo.jl is a package for creating and manipulating database queries. By pressing the touch pad, the SQL query be executed and get inserted into the database.

* Rotating a Cube in Blender with BlenderPlot.jl
Blender is a popular 3D modeling tool.
BlenderPlot.jl is a Julia package for data plotting in Blender. It worked with pyjulia and PyCall.jl.
For additional configuration to work with Push,
I added a network layer of OSC (Open Sound Control).
The OSC server has started to receive the MIDI events.
So Blender has to be controlled by Push.
Now it is possible that rotating a Cube in 3D by controlling the knobs. It's also possible to set the current frame of the Timeline. In the human walk example, when controlling the specific knob, the current frame has changed by forward/backward way that the human walk animation played plausibly.

Conclusion and Future Work

Julia is good for the hardwares and machines.
There are many fields that can be applied.
For the future work, I'd like to work with Ableton Live API to collaborate with musicians with Julia.
Machine-learning based Live coding music would be fun. Visualization with Blender would be nice too.

References/Acknowledgements

Mostly personal work. Thanks for all the starring stars of the projects.

The following sites are mentioned:
- PushInterface.jl https://github.com/wookay/PushInterface.jl
- The Ableton Push 2 MIDI and display interface manual https://github.com/Ableton/push-interface
- Palette https://palettegear.com/