MRuby-Zest - A new GUI toolkit for audio programs

Mark McCurry

June 5th, 2018

Zyn-Fusion


ZYN
FILE
MIDI
LEARN
ATTACK TIME
FINE
VOLUME
NRPN
KEY SHIFT
PART SETTINGS
VCE 1
GLOBAL
VOICE
OSC
MOD-OSC
MODULATION
VOICE LIST
RESONANCE
C
P
FREE
ADD
DELETE
SUSTAIN POINT
2
288.1 MSEC
BROWSER
MIXER
KITS
MIDI LEARN
EFFECTS
ADD
SUB
PAD
GENERAL
ENVELOPE
LFO
VOL
SCALE
PAN
STRETCH
STRENGTH
TIME
STEREO
RND GRP
A.DT
D.DT
S.VAL
R.DT
STRETCH
FRCR
LIN/LOG
FREQ
DEPTH
START
STR
DELAY
A.R.
F.R.
SINE
SYNC
AMPLITUDE
FREQUENCY
FILTER
QWERTY
VELOCITY
VRND
OCTAVE
C.VAL
01: MOD.WHEEL
MIDI CC

Another GUI Toolkit? Why?

Another GUI Toolkit? Why?

- Qt
- GTK
- AVTK
- robtk
- DPF
- JUCE
- fltk

Challenges

Toolkit's:

- Maturity
- Suitability for use in plugins
- Development speed

Zyn Timeline

- $\approx$ 3 months of time
- $\approx$ 30 views to implement

Zyn Timeline

- $\approx$ 3 months of time
- $\approx$ 30 views to implement
- Not a lot of time

Zyn Timeline

- I'm bad at GUI Programming

Zyn Timeline

- I'm bad at GUI Programming
- (and I hope I'm not alone)

Zyn Timeline

- Target DRY
- Target fast feedback loop
- Target long term maintainiability

Zyn Timeline

New framework is a reasonable investment

- From scratch look and feel
- Take advantage of Zyn metadata model
- Provide something that can be enhanced long term

Borrowing Ideas

- Qt's QML
- QML's built in hotloading
- rtosc's metadata system

Qt's QML

Provides easy way to:

- Build widget trees
- Define custom behavior for edge cases
- Constrain how you expect widgets to be extended

Qt's QML

- QML's organization is great, but javascript is not-fun
- Provides a means for organizing widgets
- Makes widget extension easy

MRuby - Not Ruby

QML before - a parsing standpoint

```
MouseArea {
    id: ma
    property var styleData

    anchors.fill: parent
    onPressed: {
        //javascript
        parent.currentRow = styleData.row
        parent.selection.select(styleData.row)
    }
    onClicked: {
        //javascript
        console.log(styleData.value)
    }
}
```

QML after - a parsing standpoint

```
MouseArea {
    id: ma
    property var styleData

    anchors.fill: parent
    onPressed: lambda {
        #Ruby
        parent.currentRow = styleData.row
        parent.selection.select styleData.row
    }
    onClicked: lambda {
        #Ruby
        puts styleData.value.inspect
    }
}
```

So What is MRuby-Zest

- Uses QML's syntax in a MRuby environment
- Builds off of rtosc's exported metadata for quick dev

So What is MRuby-Zest

- Uses QML's syntax in a MRuby environment
- Builds off of rtosc's exported metadata for quick dev
- $\approx 1,000$ commits so far
- $\approx$ 15 kloc of QML (widgets)
- $\approx$ 6 kloc ruby
- $\approx$ 7 kloc C

Components of the toolkit

- mruby-qml-parse
- mruby-qml-spawn

Components of the toolkit

- mruby-qml-parse
- mruby-qml-spawn
- mruby-zest

Components of the toolkit

- mruby-qml-parse
- mruby-qml-spawn
- mruby-zest
- osc-bridge

Components of the toolkit

- mruby-qml-parse
- mruby-qml-spawn
- mruby-zest
- osc-bridge
- mruby-widget-lib

OSC bridge

```
{
    "path"     : "/part[0,15]/kit[0,15]/padpars/GlobalFilter/basefreq",
    "shortname": "cutoff",
    "name"     : "basefreq",
    "tooltip"  : "Base cutoff frequency",
    "units"    : "Hz",
    "scale"    : "logarithmic",
    "type"     : "f",
    "range"    : [31.25,32000]
},
{
    "path"     : "/part[0,15]/kit[0,15]/padpars/GlobalFilter/freqtracking",
    "shortname": "f.track",
    "name"     : "freqtracking",
    "tooltip"  : "Frequency Tracking amount",
    "units"    : "%",
    "scale"    : "linear",
    "type"     : "f",
    "range"    : [-100,100],
    "default"  : "0.0f"

},
```

QML Loading

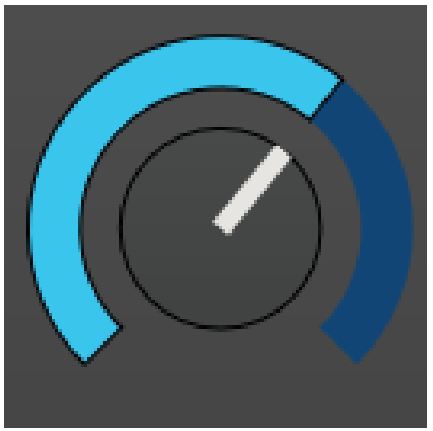
- Class definitions
- Property definitions
- Method definitions
- Class instance specialization**

QML Loading - Live or at build

- Classes can be parsed and turned into .rb at build
- .qml files can be reloaded and re-instantiated at runtime

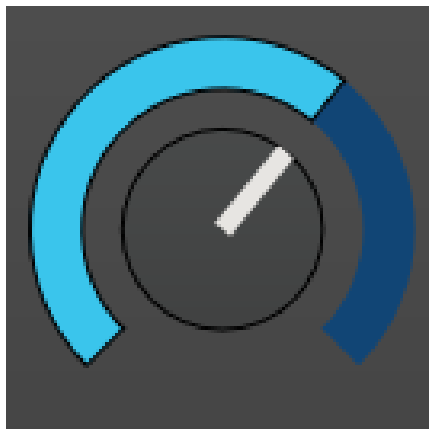
Hotloading - setup

```
Knob {
    function draw(vg) {
        active_color = :blue

        draw_outline()
        draw_xxx()
        ...
        ...
        ...
        ...
    }
}
```



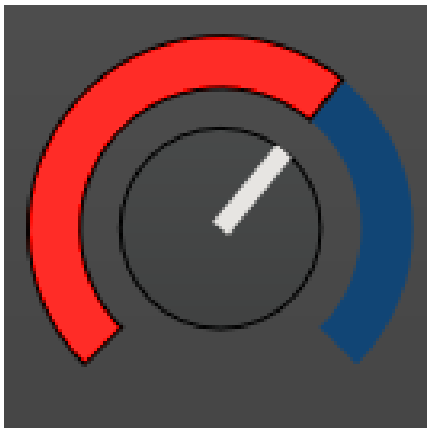
Hotloading - making changes

```
Knob {
    function draw(vg) {
        active_color = :red

        draw_outline()
        draw_xxx()
        ...
        ...
        ...
        ...
    }
}
```



Hotloading - update on saving

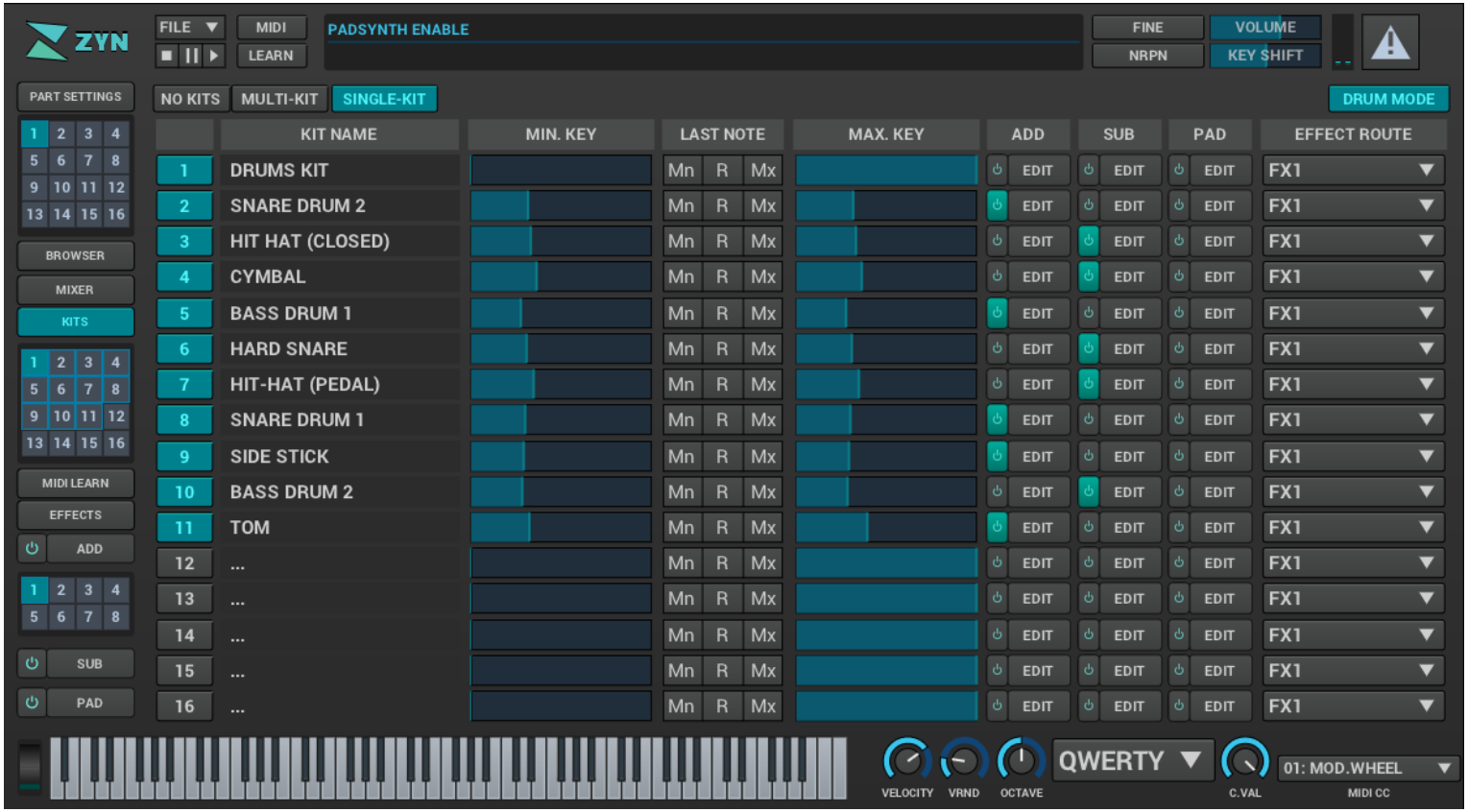
```
Knob {
    function draw(vg) {
        active_color = :red

        draw_outline()
        draw_xxx()
        ...
        ...
        ...
        ...
    }
}
```



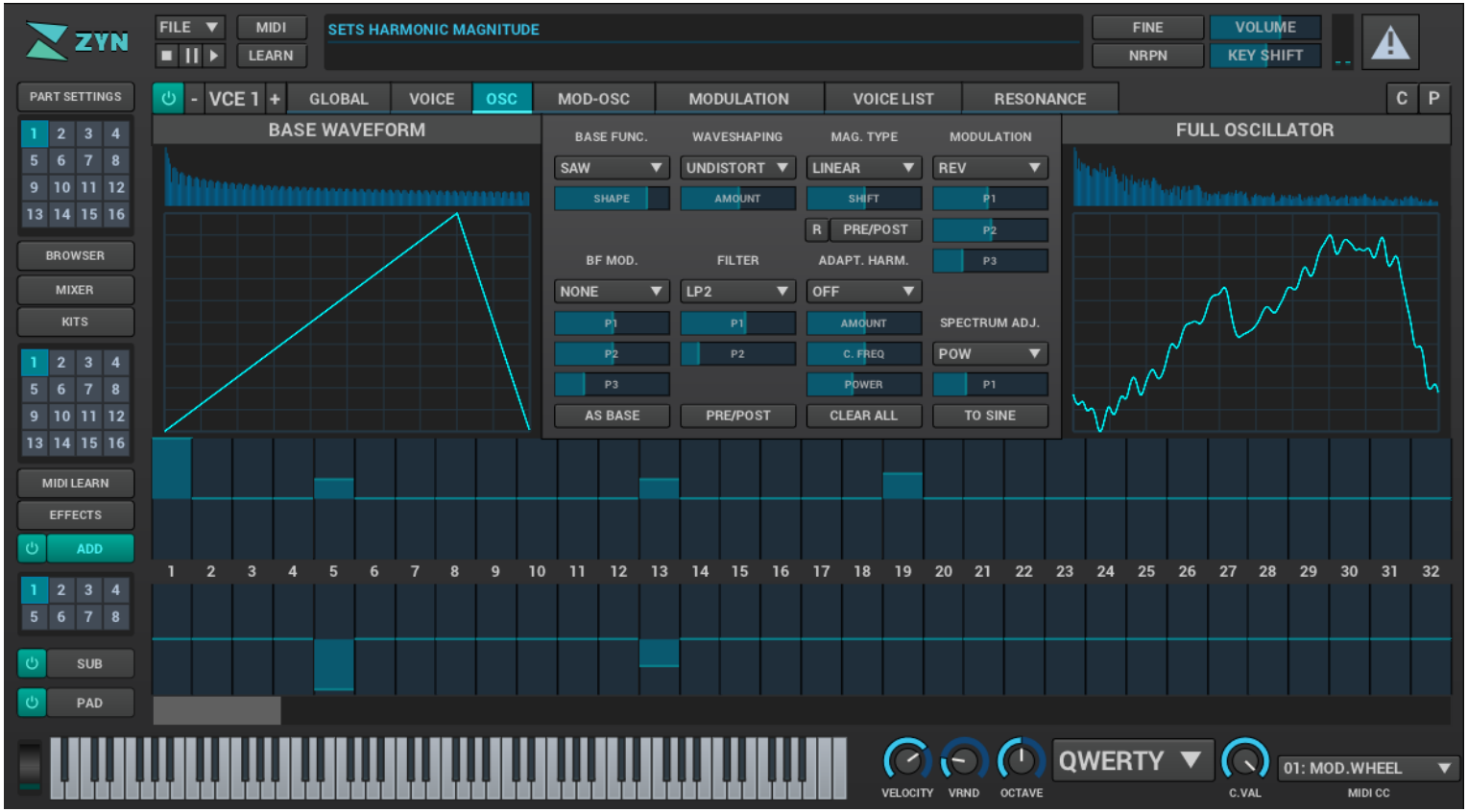
Widgets


ZYN
FILE
MIDI
LEARN
ATTACK TIME
FINE
NRPN
VOLUME
KEY SHIFT
PART SETTINGS
VCE 1
GLOBAL
VOICE
OSC
MOD-OSC
MODULATION
VOICE LIST
RESONANCE
C
P
FREE
ADD
DELETE
SUSTAIN POINT
2
288.1 MSEC
BROWSER
MIXER
KITS
MIDI LEARN
EFFECTS
ADD
SUB
PAD
GENERAL
ENVELOPE
LFO
VOL
SCALE
PAN
STRETCH
STRENGTH
TIME
STEREO
RND GRP
A.DT
D.DT
S.VAL
R.DT
STRETCH
FRCR
LIN/LOG
FREQ
DEPTH
START
STR
DELAY
A.R.
F.R.
SINE
SYNC
AMPLITUDE
FREQUENCY
FILTER
QWERTY
VELOCITY
VRND
OCTAVE
C.VAL
01: MOD.WHEEL
MIDI CC

Widgets, Widgets



Widgets, Widgets, Widgets



Even more widgets

Future Work

- Translations
- More data visualizations
- Animations
- Automated Screenshot collection
- Exploiting the scripting capiabilities more
- Separation from Zyn

Conclusions

- MRuby-Zest powers Zyn-Fusion
- Adds hotloading and scripting to the plugin level UI design
- Builds off existing tools for streamlined dev
- It's new and ready to adapt

Questions?