Web Audio Modules

Jari Kleimola
Dept. of Computer Science
Aalto University
Finland

Oliver Larkin
Music Department
University of York
UK
outline

background
WAM API
demo
implementations
conclusion
intro

browsers
web audio modules
web browser affordances

<table>
<thead>
<tr>
<th></th>
<th>DAWs</th>
</tr>
</thead>
<tbody>
<tr>
<td>plugins</td>
<td></td>
</tr>
<tr>
<td>music making, performing</td>
<td></td>
</tr>
</tbody>
</table>

JavaScript
WAM API
JavaScript (← C++)

web page

plugin (DSP)
native plugin apis

- vST
- RTAS
- AAX
- vST
- Audio Units
- LV2
- LADSPA
- DSSI
- JUCE
DAW Plugins for Web Browsers

Jari Kleimola / WAC 2015

https://mediatech.aalto.fi/publications/webservices/dawplugins
web audio

web audio api

AudioNode → AudioNode → AudioNode

cross-compilation

C/C++ → Emscripten → asm.js

web audio modules

AudioNode → WAM → AudioNode

Web Audio Modules
Web Audio Modules

COMMUNITY SITE

synthesizers and audio effects processors for web browsers

BROWSE SYNTHS or scroll down to learn more
include into a webpage

```
<link rel="import" href="http://webaudiomodules.org/synths/wam-webdx7.html"/>
<wam-webdx7 autoconnect></wam-webdx7>
...
var dx7 = document.querySelector("wam-webdx7").controller;
dx7.postMidi([...]) etc.
```

create one yourself

- inherit WAM.Controller
- inherit WAM.Processor / WAM.Synth
- or (re-)use C++ code
webCZ-101/VirtualCZ
IPlug Web Target

- Export IPlug C++ code to ASM.js with template controller JS + HTML glue
- Currently plug-in GUI needs to be re-written
virtual Yamaha DX7 synthesizer
   - sound engine by Raph Levien
   - ported from vanilla C++, GUI took more time than DSP

patchcloud
   - lots of patches in the internet
   - interactive WebGL visualization
### Latency

<table>
<thead>
<tr>
<th>WAM</th>
<th>default</th>
<th>buffer = 32</th>
</tr>
</thead>
<tbody>
<tr>
<td>webCZ-101</td>
<td>48 ms</td>
<td>33 ms</td>
</tr>
<tr>
<td>webDX7</td>
<td>45 ms</td>
<td>31 ms</td>
</tr>
</tbody>
</table>

### Polyphony

<table>
<thead>
<tr>
<th>WAM</th>
<th>asm.js</th>
<th>native</th>
<th>Polyphony</th>
</tr>
</thead>
<tbody>
<tr>
<td>webCZ-101</td>
<td>60</td>
<td>200</td>
<td>30%</td>
</tr>
<tr>
<td>webDX7</td>
<td>17</td>
<td>128</td>
<td>13%</td>
</tr>
</tbody>
</table>
commercial issues

- Relatively easy to add support for WAM API to an existing C++ codebase
  - Multithreading, explicit SIMD and platform specific ASM might cause problems
  - GUI needs reworking (but Web UI is the way interfaces are going anyway)
  - Performance slower than native

- emscripten’s ASM.js code browseable, and algorithms viewable
- Wams are easily shareable
future work

hosting in soundation
PNaCl + Web Assembly
AudioWorkerNode
GUI codebase reuse

http://chrome.soundation.com
conclusion

proposed streamlined Web Audio Modules API
  – virtual instruments and audio effects processors for web browsers
  – load with the rest of the page (no manual installation)

two proof of concept implementations
  – latencies higher than in native, but gap is narrowing

community website
  – hosts WAMs
  – documentation
  – feedback
THANKS FOR LISTENING!

webaudiomodules.org

jari.kleimola@alumni.aalto.fi
oliver.larkin@york.ac.uk