Porting Wwise to the web

June 28, 2019
What is Wwise?

Authoring Tool

Sound Banks

Game + Sound Engine
Why is Wwise important?
Attempt #1: Just make it play a sound!

Hurdles:
- Architecture that is neither x86 nor ARM
- SSE emulation did not work
- Atomics: some compiler intrinsics were not supported
- pthreads implementation is a leaky abstraction
Attempt #1: Deadlock galore

- Threads don’t actually start before `main()` returns
- Only the main thread can initialize SDL2
- SDL2 Audio is implemented using ScriptProcessorNode… AudioThread==Main thread!
Attempt #2: Enter AudioWorklet

IntegrationDemo.js

Game + Sound Engine (WASM)

IntegrationDemo.worker.js (WorkerGlobalScope)
IntegrationDemo.processor.js (AudioWorkletGlobalScope)

IntegrationDemo.js

Game + Sound Engine (WASM)

IntegrationDemo.worker.js (WorkerGlobalScope)
IntegrationDemo.processor.js (AudioWorkletGlobalScope)

import
import

postMessage(wasmHeap)
In retrospect

The good:
- AudioWorklet: a major improvement
- Didn’t have actual audio issues, just needed to learn the APIs

The bad:
- Threading is difficult to get right
- Debugging “experience”
- Too far from the hardware

The ugly:
- SharedArrayBuffer’s future seems uncertain
- Slow adoption by browsers other than Chrome
- What about mobile browsers?
Thank you!