Agenda

1. Quick introduction to SOUL
2. Getting your machines setup
3. Let's write a synth
4. Q/A

What is SOUL?

- SOUL is a new Domain Specific Language (DSL) for writing the real-time parts of an audio algorithm
- The language is designed to be JIT compiled to run on heterogeneous CPUs and DSPs
- SOUL includes a run-time platform to execute the code, both locally and remotely

Get your machines set up for coding

Download tooling from GitHub

https://github.com/soul-lang/SOUL

```
$ curl -L https://github.com/soul-lang/SOUL/releases/download/0.8.830/
binaries-osx-x64.zip --output binaries-osx-x64.zip
$ unzip binaries-osx-x64.zip
$ alias soul=$PWD/osx/x64/soul
```

Get your machines set up for coding

Install/Setup VS Code with SOUL syntax highlighting

- Install VS Code https://code.visualstudio.com/
- Install the VS Code extension for SOUL
  (see SOUL/tools/editors/vs_code_extension)

Follow the examples on soul-dev

The steps can be reached at:

https://soul.dev/lab?id=PAW-1

https://soul.dev/lab?id=PAW-2

https://soul.dev/lab?id=PAW-3

https://soul.dev/lab?id=PAW-4

Let’s write a synth!

1. Create a new soulpatch

Use the soul command to create a new instrument soulpatch

```
$ soul create MonoSynth --synth –output=MonoSynth
```

2. Try playing it

```
$ soul play MonoSynth/MonoSynth.soulpatch
```

Let's write a synth!

3. Add a waveshaper

- Waveshapers transform input to output samples using a transfer function

  y = f(x(t))

- We are going to use the function tanh()
- We will provide a drive parameter to control the depth of the effect

Let's write a synth!

4. Adding the waveshaper to the audio graph

- connections only defined in graphs
- connect inputs to output of the same type
- graphs form directed graphs*

Let's write a synth!

5. Add an envelope

- Let's create an Attack/Release envelope
- Linear Attack
- Exponential Release
- Remove envelope from oscillator
- Connect the envelope after the waveshaper

Let's write a synth!

6. Further Enhancements

- Aliasing problems - apply an oversampling factor to the waveshaper
- Zipper noise - add slewRates for stream parameters
- Consider restructuring to create a voice
- Add polyphony

Q/A