
ADC21
HYBRID PROTOTYPING
WITH WEB TECH AND JUCE/C++
ARTHUR CARABOTT


putput


120
Done With You
Tweak
hooked
Deep Sweep
Ambience
Phase Mod
Pumping
Pitch
A♭ Maj


MOTU

How to use web technologies for productive prototyping.

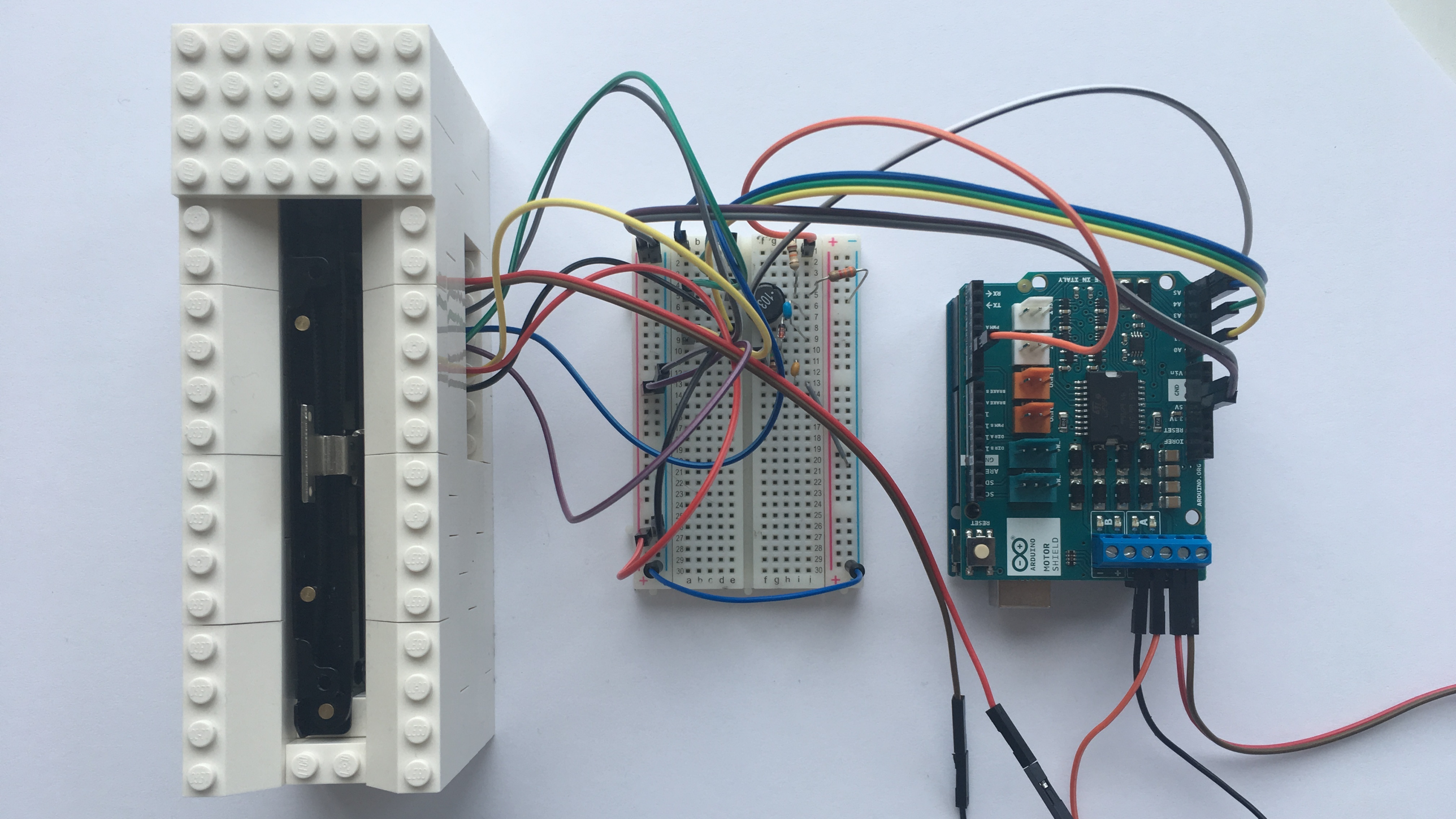
1. Prototyping

2. Why Web Tech?

3. Case Study: Arpeggiator

4. C++ / WebViews


EVERYTHING IS
PRETTY
GOOD
AKAI MPKmini
Oculus


ARDUINO
MOTOR
SHIELD
ARDUINO.ORG

1. Communication

Feedback loop with yourself

Feedback loop with others

2. Cost

Time and money

Motivation and creativity

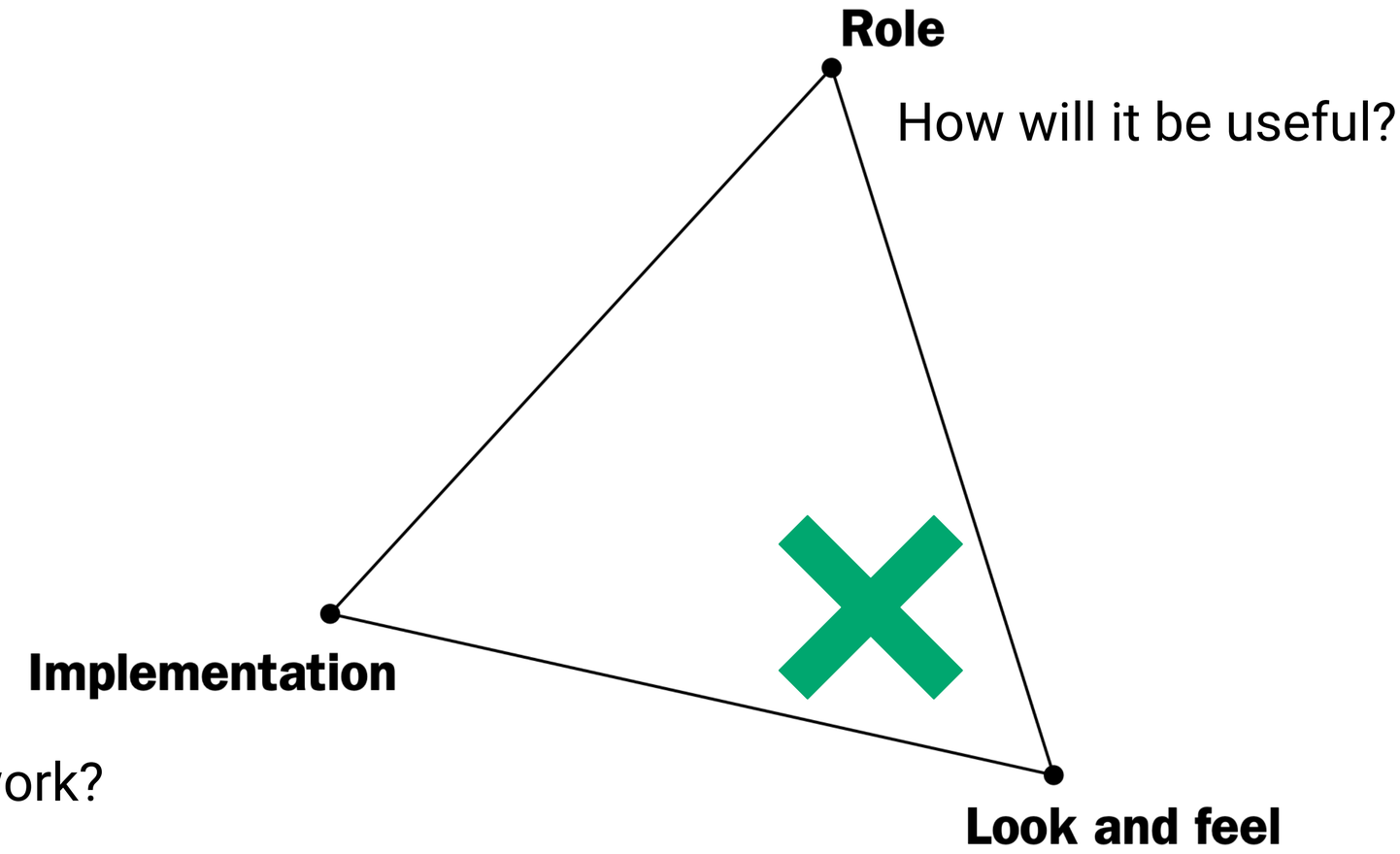
Speed

Decrease iteration time

Increase quality

Look and feel prototypes

Prototyping

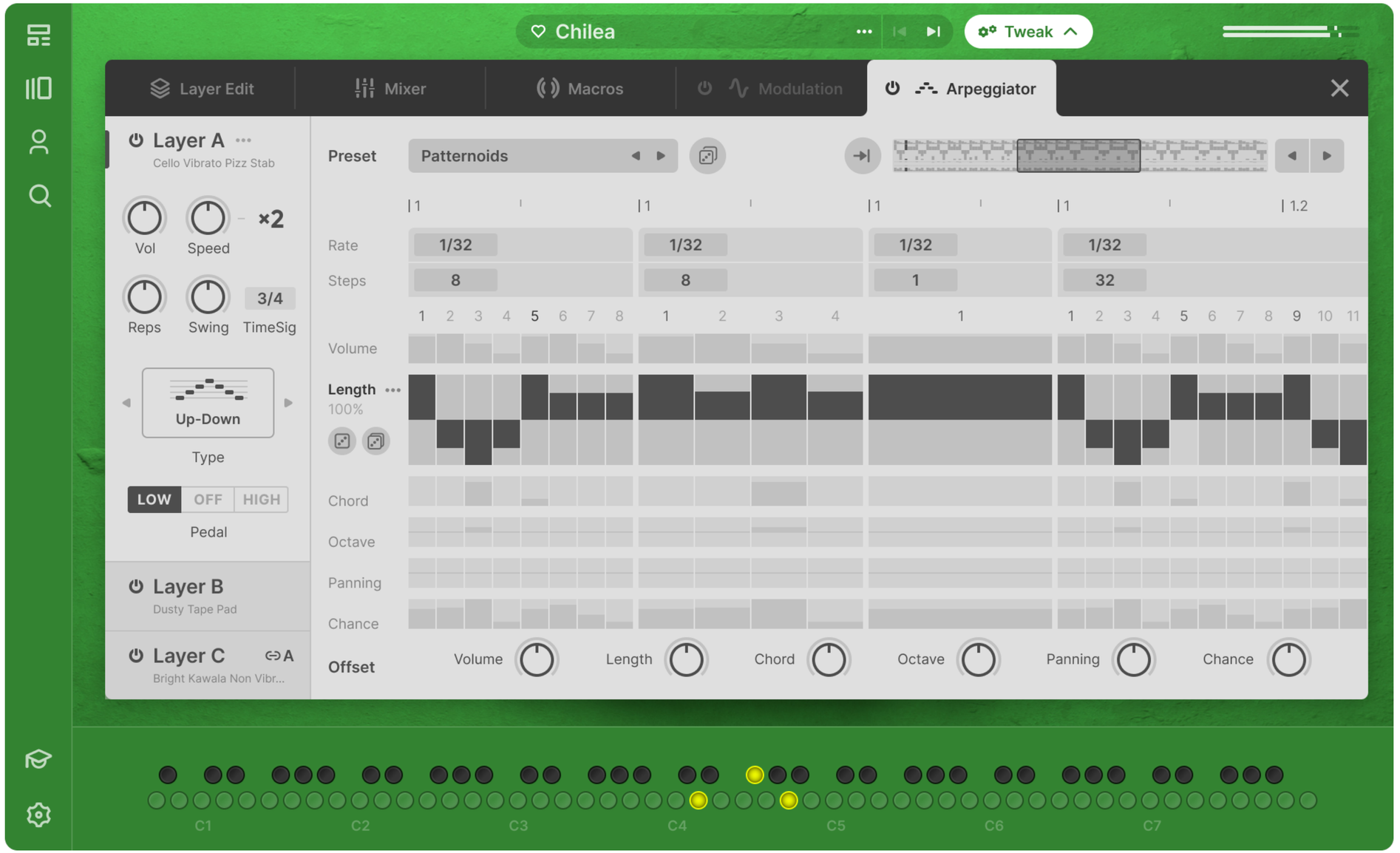


What will the experience be?

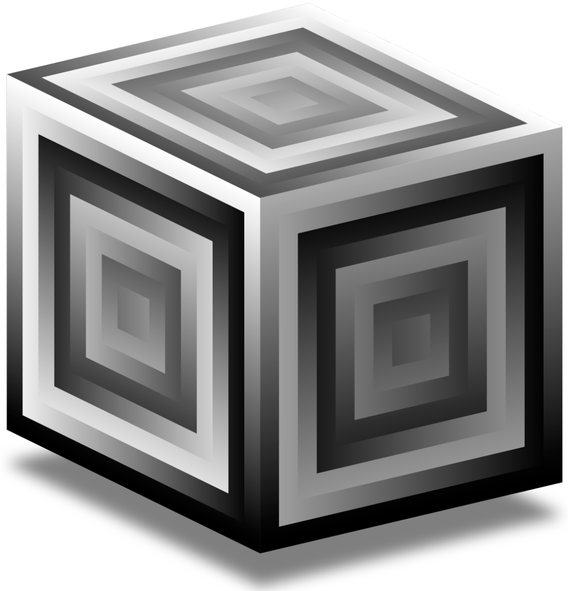
https://hci.stanford.edu/courses/cs247/2012/readings/WhatDoPrototypesPrototype.pdf

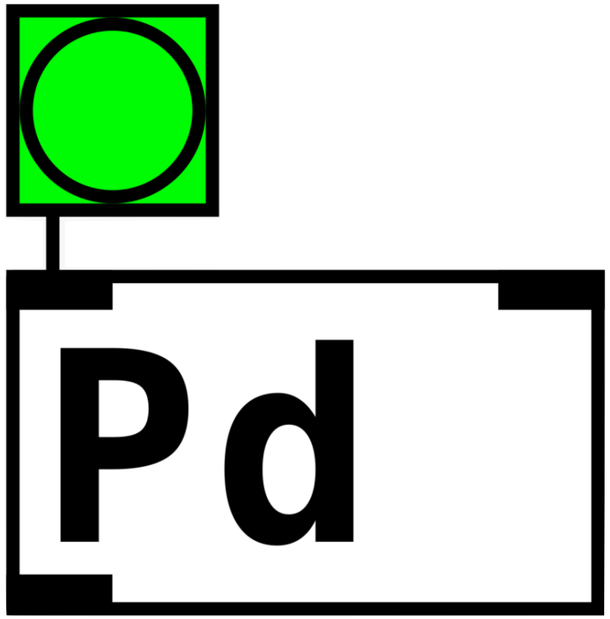
Prototyping

Prototyping

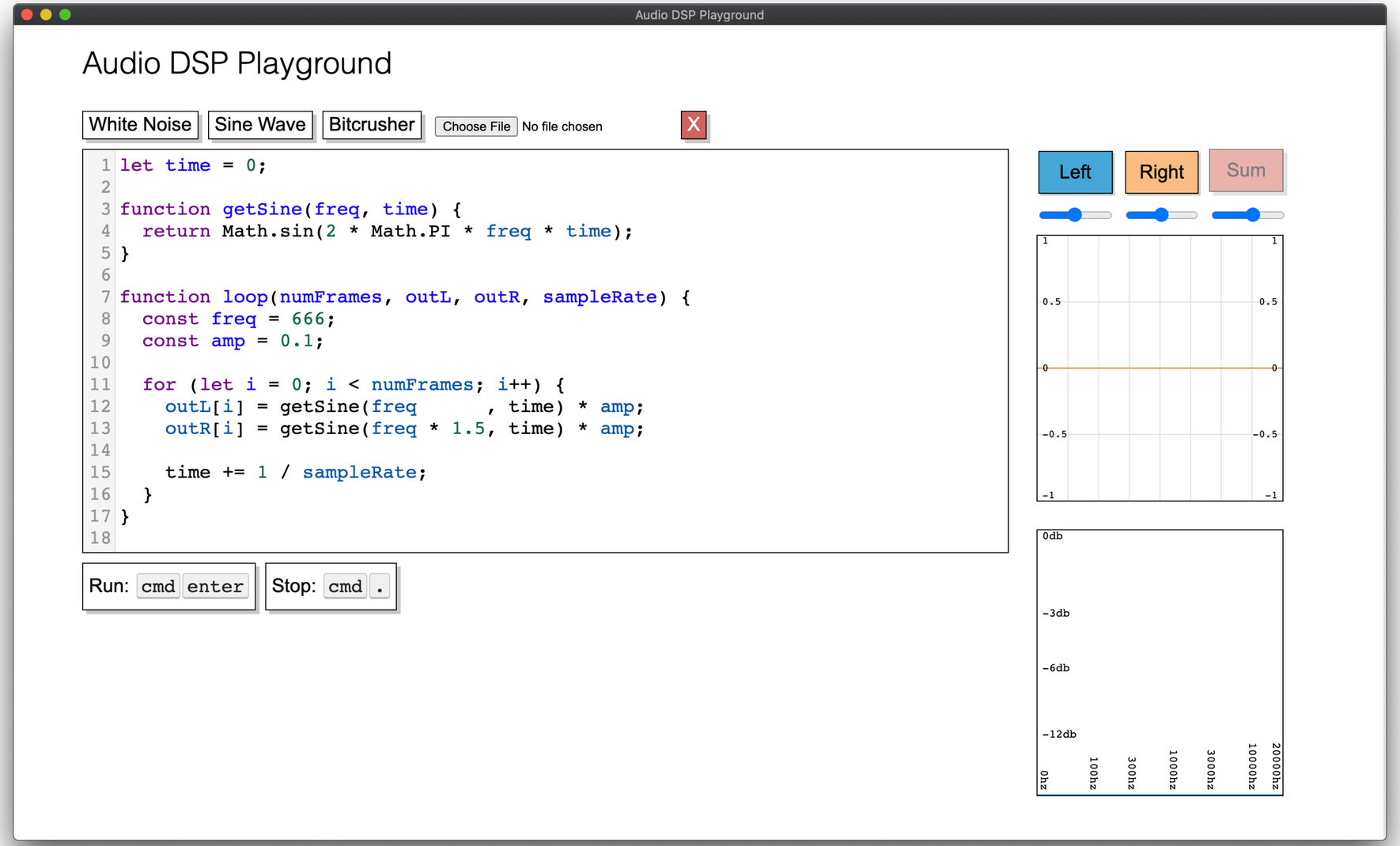


Prototyping





Prototyping

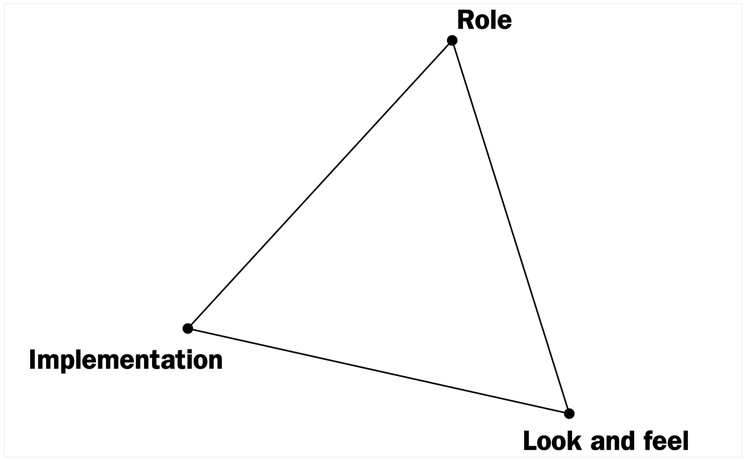


Prototyping

Marek Bereza

How to prototype Audio Software

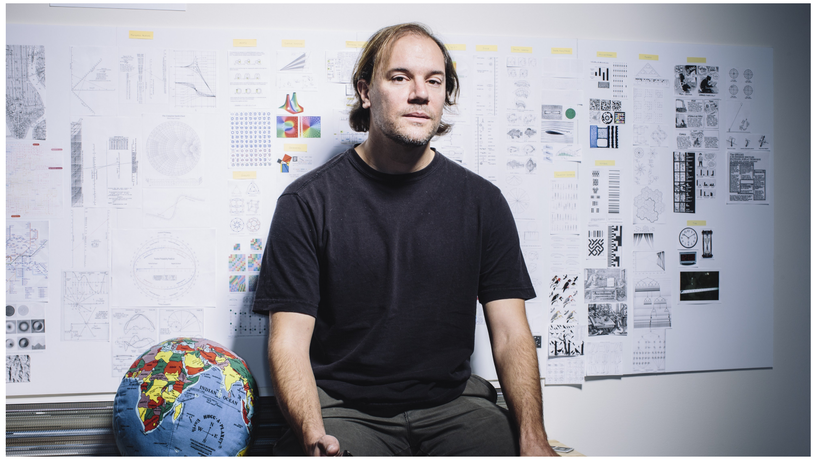
https://youtube.com/watch?v=AO1V8er6IjM



Stephanie Houde and Charles Hill

What do Prototypes Prototype?

https://hci.stanford.edu/courses/cs247/2012/readings/WhatDoPrototypesPrototype.pdf



Bret Victor

Inventing on Principle

https://youtube.com/watch?v=a-OyoVcbwWE

1. Prototyping

2. Why Web Tech?

3. Case Study: Arpeggiator

4. C++ / WebViews

1. **You**
2. Values of your tools
3. Things to leverage

Be good

1. You
2. **Values of your tools**
3. Things to leverage

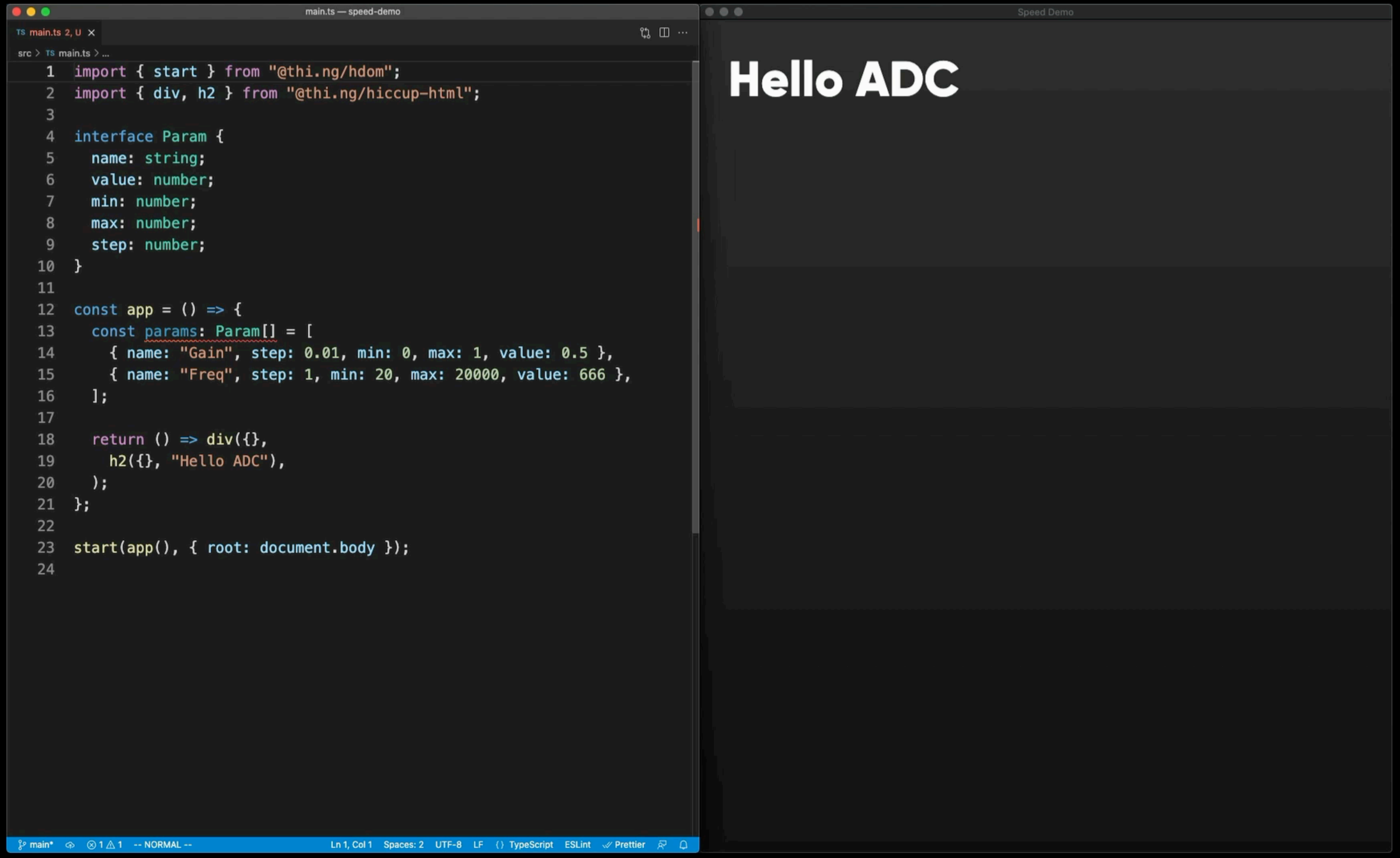
“The limits of my language
mean the limits of my world.”

Ludwig Wittgenstein
Tractatus Logico-Philosophicus, 1922

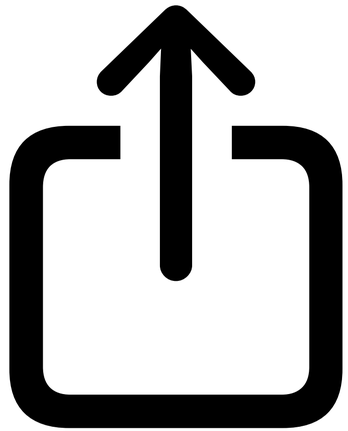
Why Web Tech?

JUCE

Speed


main.ts — speed-demo
main.ts 2, U
src > main.ts > ...
import { start } from "@thi.ng/hdom";
import { div, h2 } from "@thi.ng/hiccup-html";
interface Param {
name: string;
value: number;
min: number;
max: number;
step: number;
}
const app = () => {
const params: Param[] = [
{ name: "Gain", step: 0.01, min: 0, max: 1, value: 0.5 },
{ name: "Freq", step: 1, min: 20, max: 20000, value: 666 },
];
return () => div({},
h2({}, "Hello ADC"),
);
};
start(app(), { root: document.body });
main*
1 1
-- NORMAL --
Ln 1, Col 1
Spaces: 2
UTF-8
LF
TypeScript
ESLint
Prettier
Speed Demo
Hello ADC

Why Web Tech?



“If I send people an executable, they never open it. If I send them a link, they have no excuse.”

Alex Warth

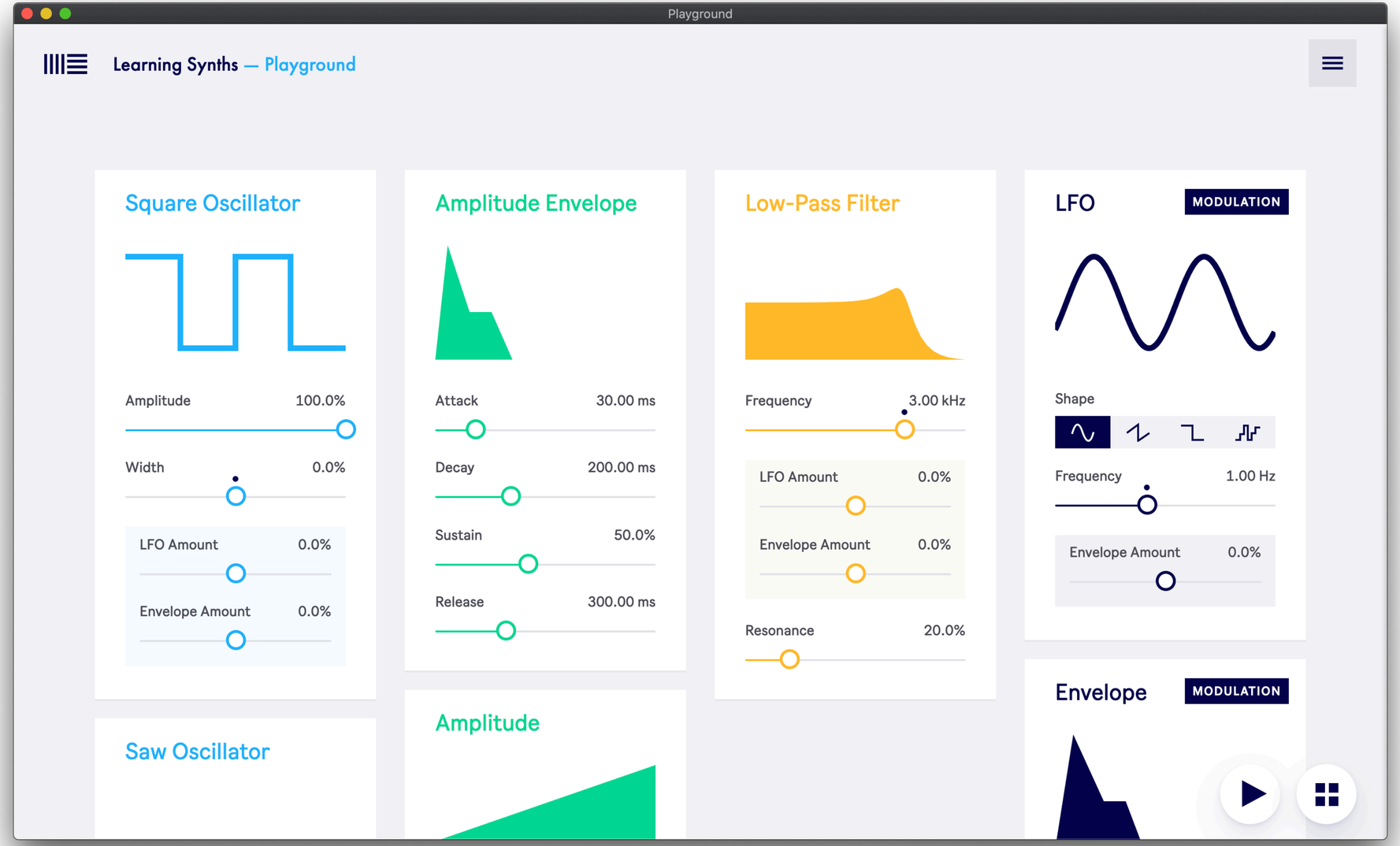
1. You
2. Values of your tools
3. **Things to leverage**

Web Audio API

Web MIDI API

Canvas / WebGL

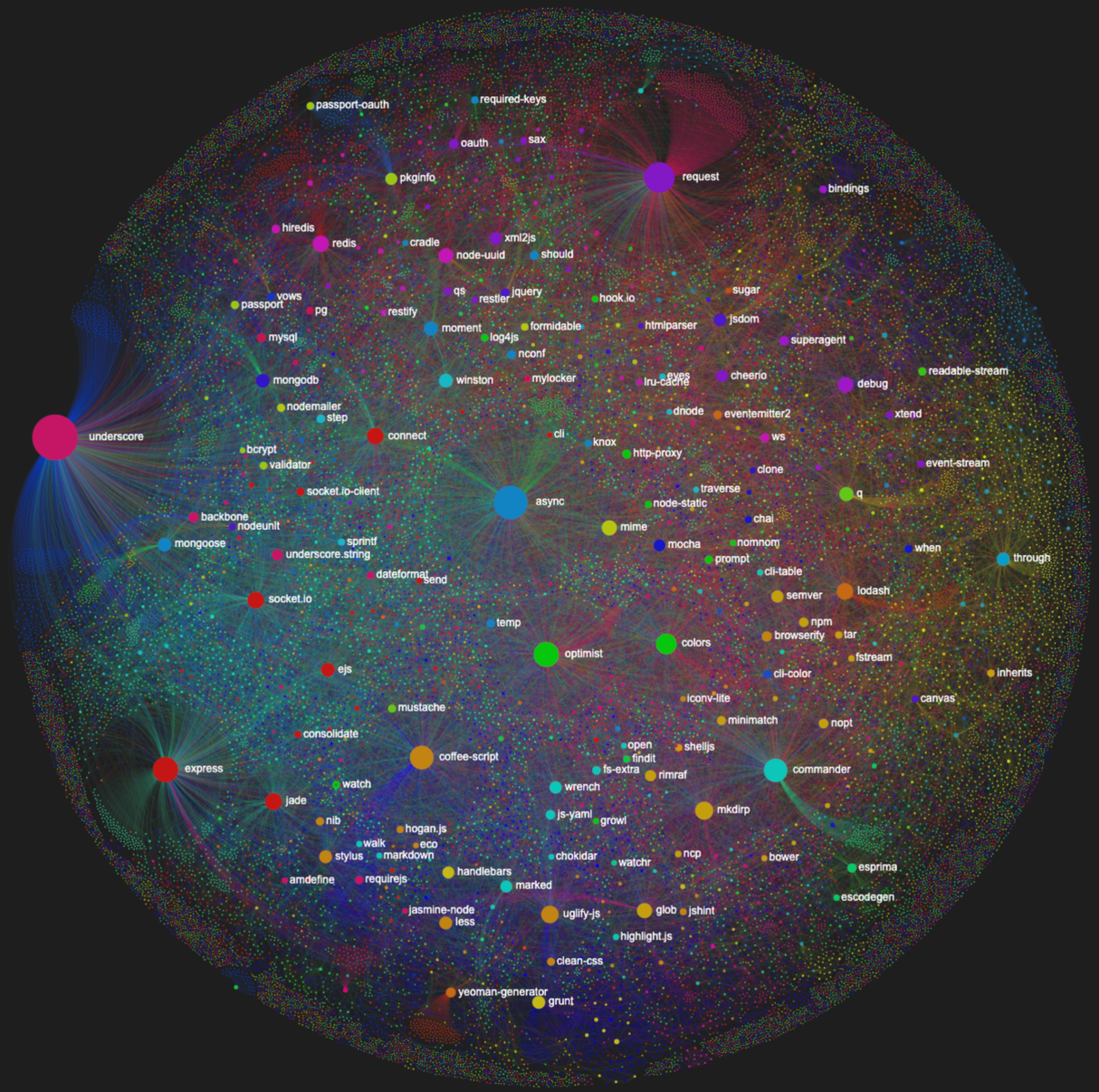
Why Web Tech?



https://learningsynths.ableton.com

Why Web Tech?

Your bag of tricks



Production vs Prototyping

Why Web Tech?

“Firstly, programs developed mainly to demonstrate feasibility may turn out in the long term to be **difficult to maintain and develop**; and secondly, their temporary user interfaces **may never be properly redesigned** before the final system is released.”

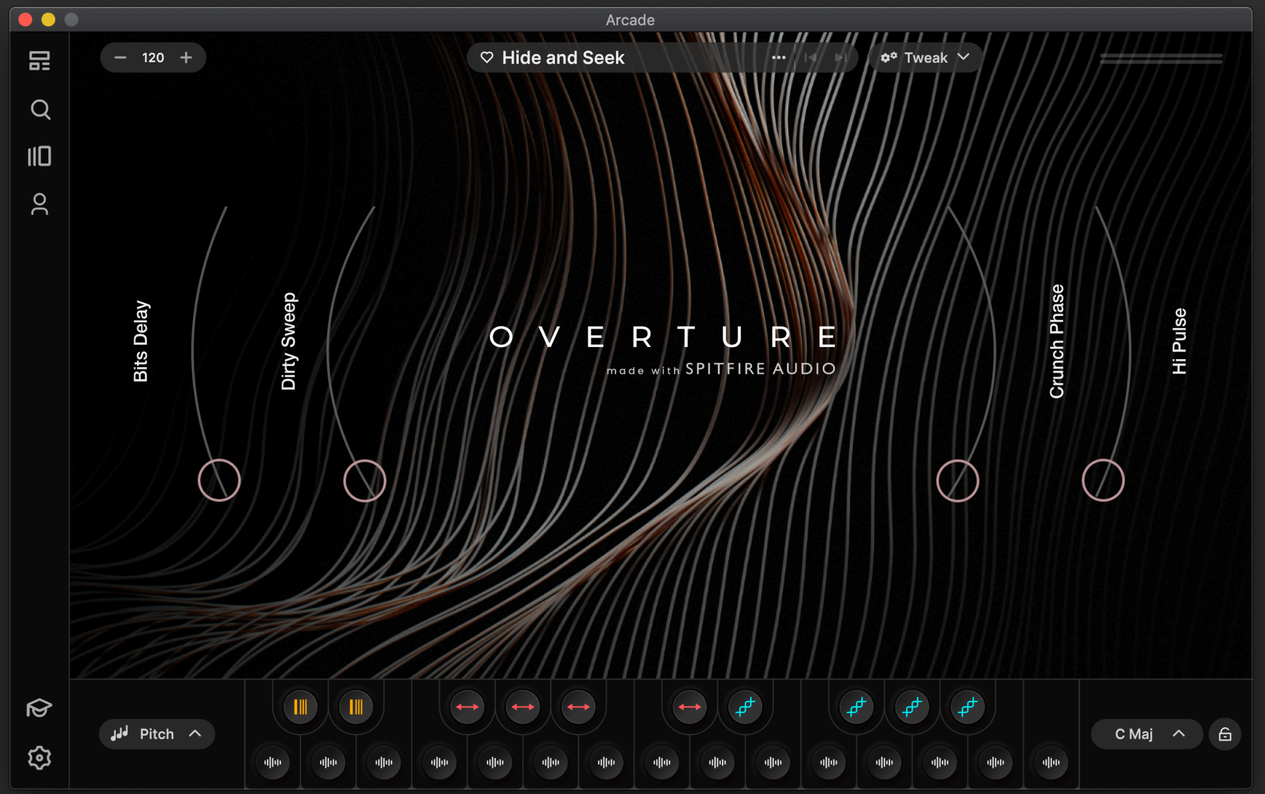
Stephanie Houde and Charles Hill
What do Prototypes Prototype? 1997

“There’s nothing more permanent than a temporary solution.”

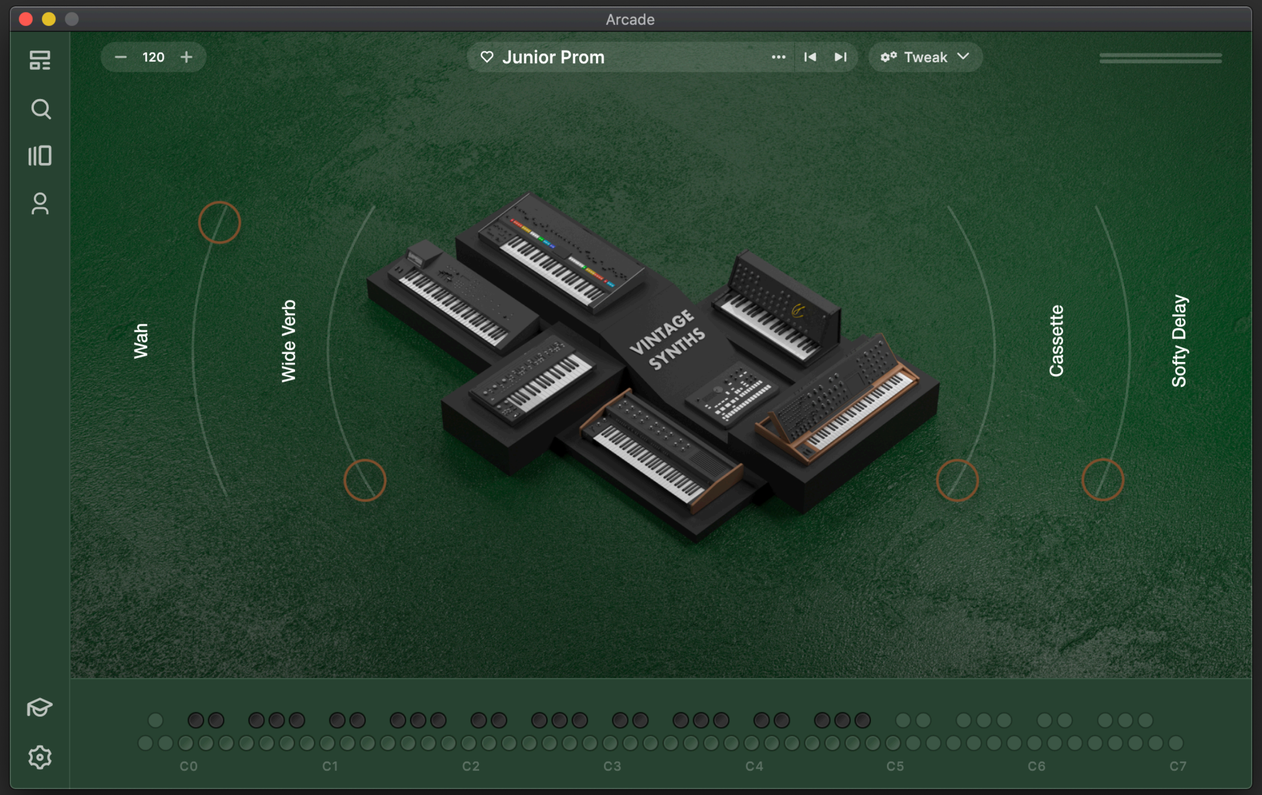
Amit Shoham

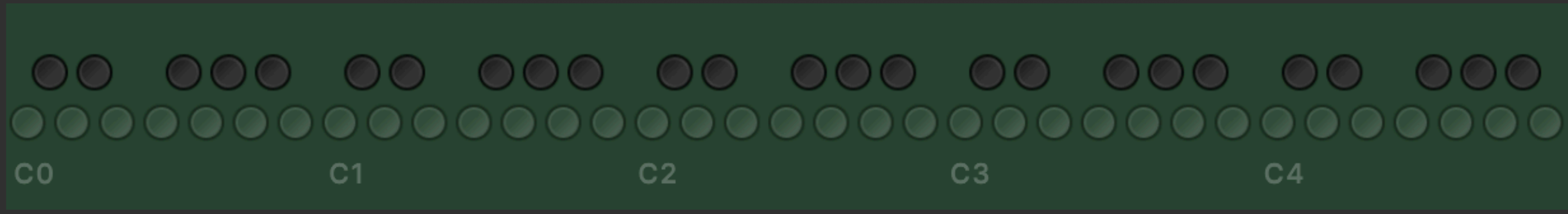
1. Prototyping
2. Why Web Tech?
3. Case Study: Arpeggiator
4. C++ / WebViews

Case Study: Arpeggiator



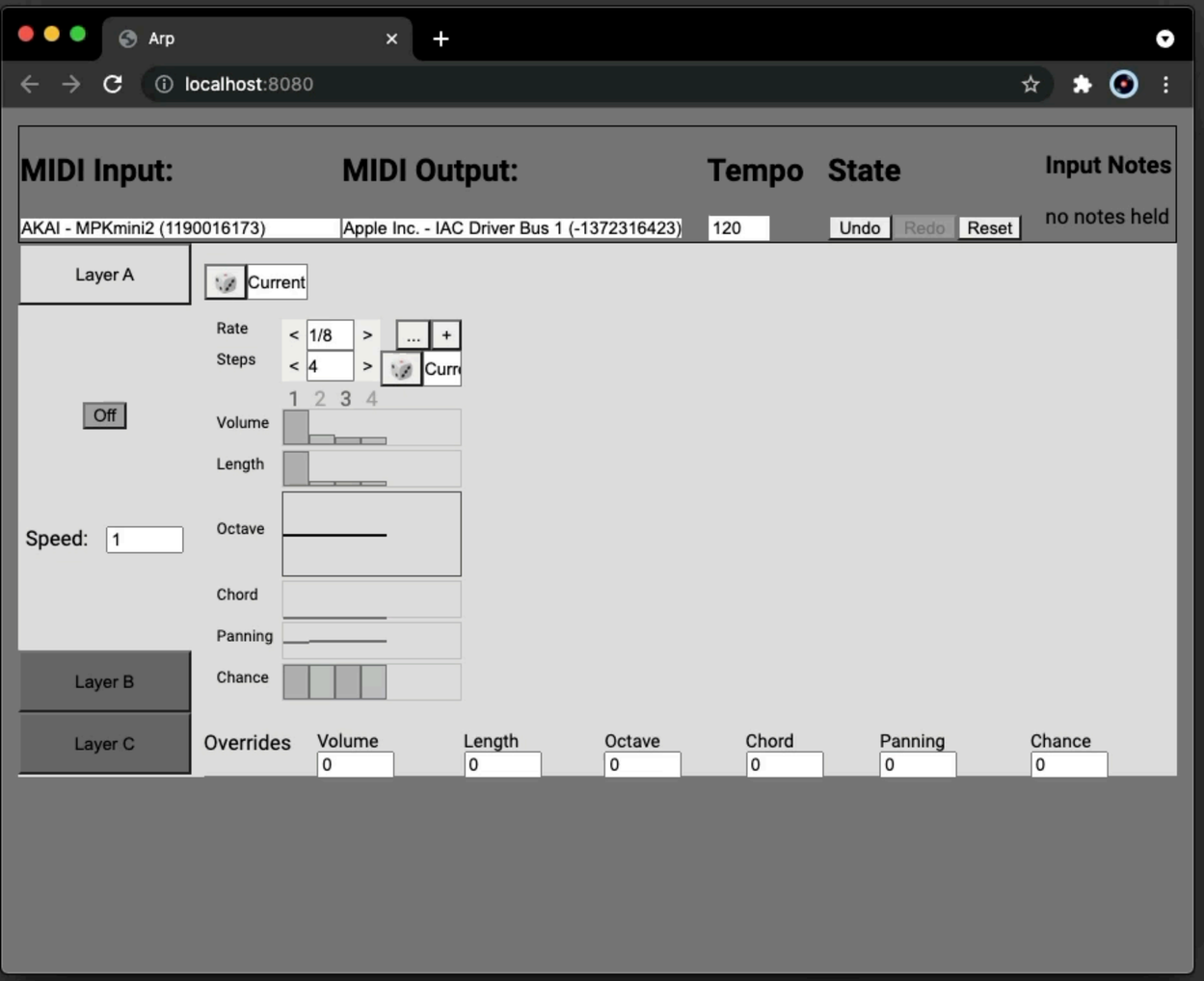
Case Study: Arpeggiator





Case Study: Arpeggiator

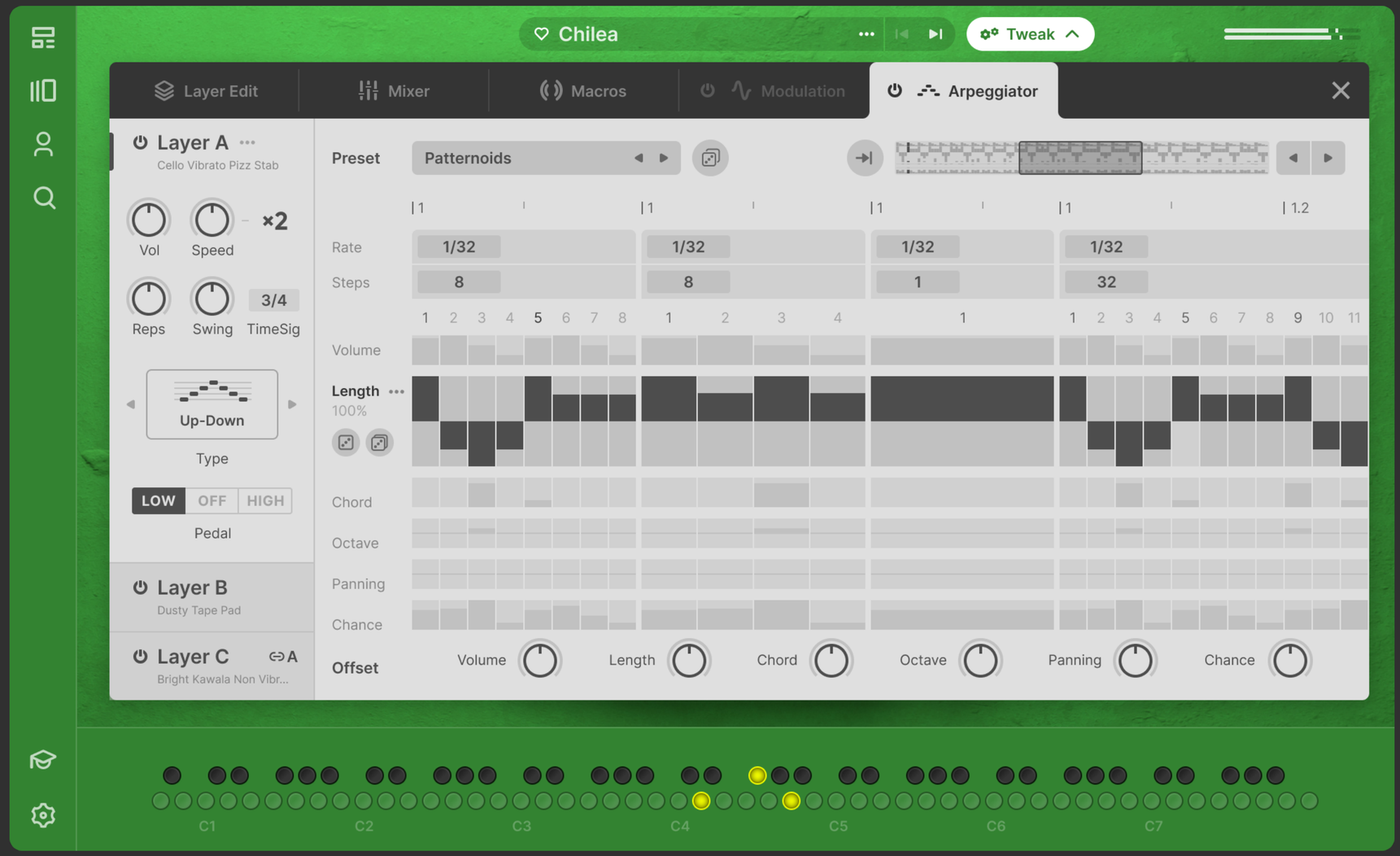
Case Study: Arpeggiator


Arp
localhost:8080
MIDI Input:
MIDI Output:
Tempo
State
Input Notes
AKAI - MPKmini2 (1190016173)
Apple Inc. - IAC Driver Bus 1 (-1372316423)
120
Undo
Redo
Reset
no notes held
Layer A
Current
Rate
1/8
Steps
4
1 2 3 4
Off
Volume
Length
Octave
Chord
Panning
Chance
Speed:
1
Layer B
Layer C
Overrides
Volume
Length
Octave
Chord
Panning
Chance
0


Arcade
120
Space Dust
Tweak
Space
Sprinkles
Bandpass
Width
BRAIN WAVES
C0
C1
C2
C3
C4
C5
C6
C7


MIDI Studio
Default
MPKmini2
IAC Driver

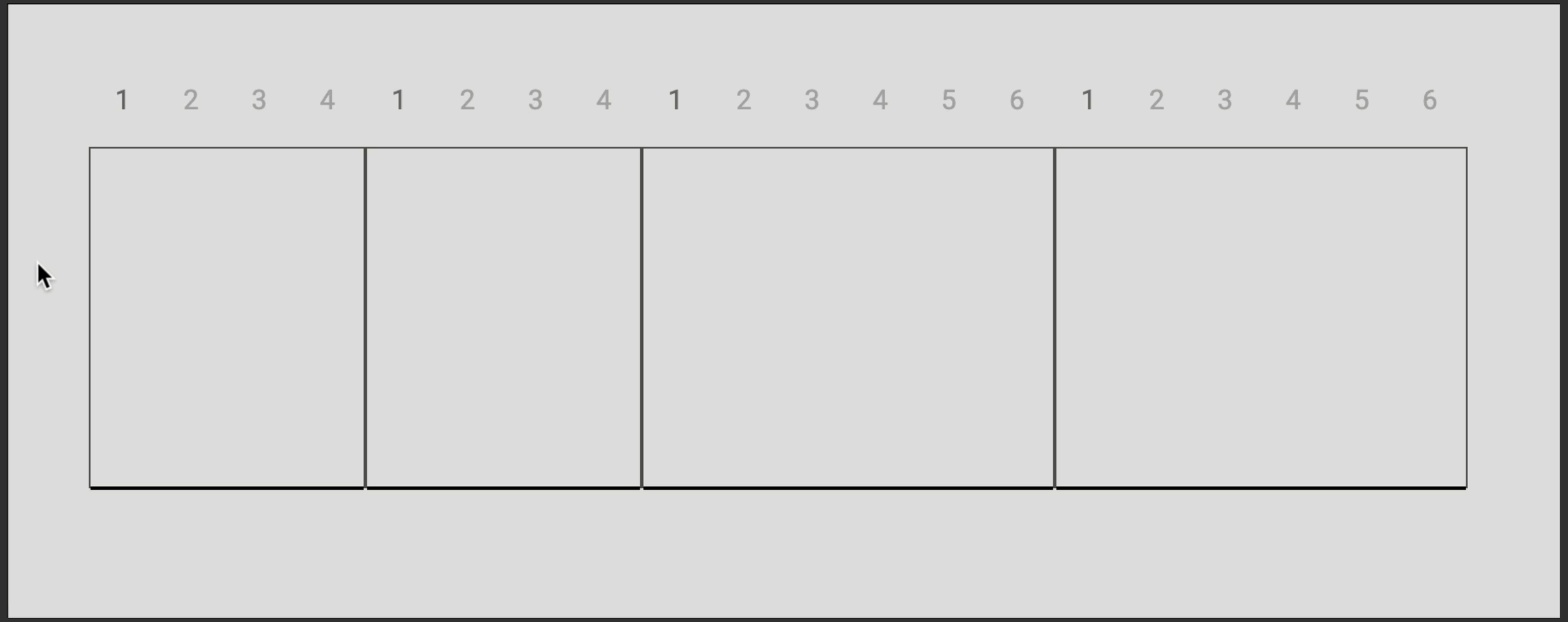
Case Study: Arpeggiator



Case Study: Arpeggiator

Case Study: Arpeggiator

Case Study: Arpeggiator



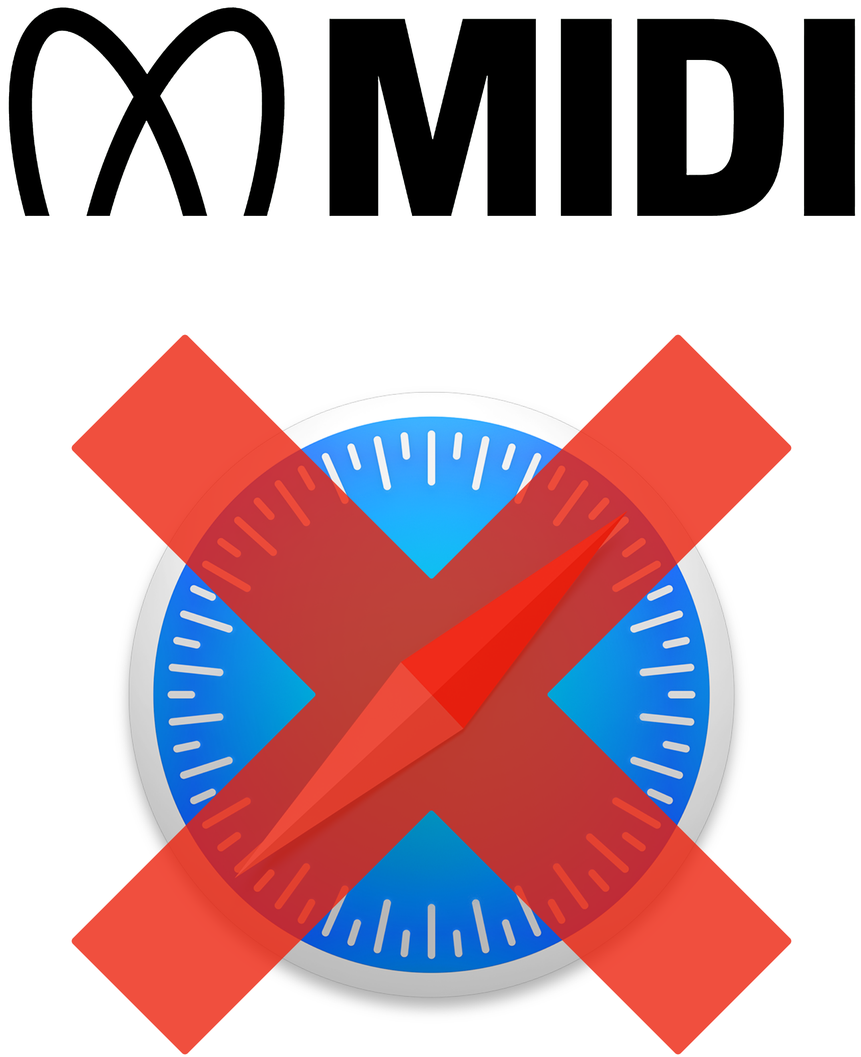
Case Study: Arpeggiator

```ts
export interface IStep {
  volume_01: number;
  length_01: number;
  octave: number;
  chordProbability_01: number;
  panning_11: number;
  chance_01: number;
}

export interface IChunk {
  rate: number;
  numActiveSteps: number;
  steps: IStep[];
}

export interface ISequence {
  chunks: IChunk[];
}
```

“This thing is amazing and not like anything else out there. It has everything I want in an arp mode.”

Nestor Estrada - Sound Designer, Output

Case Study: Arpeggiator

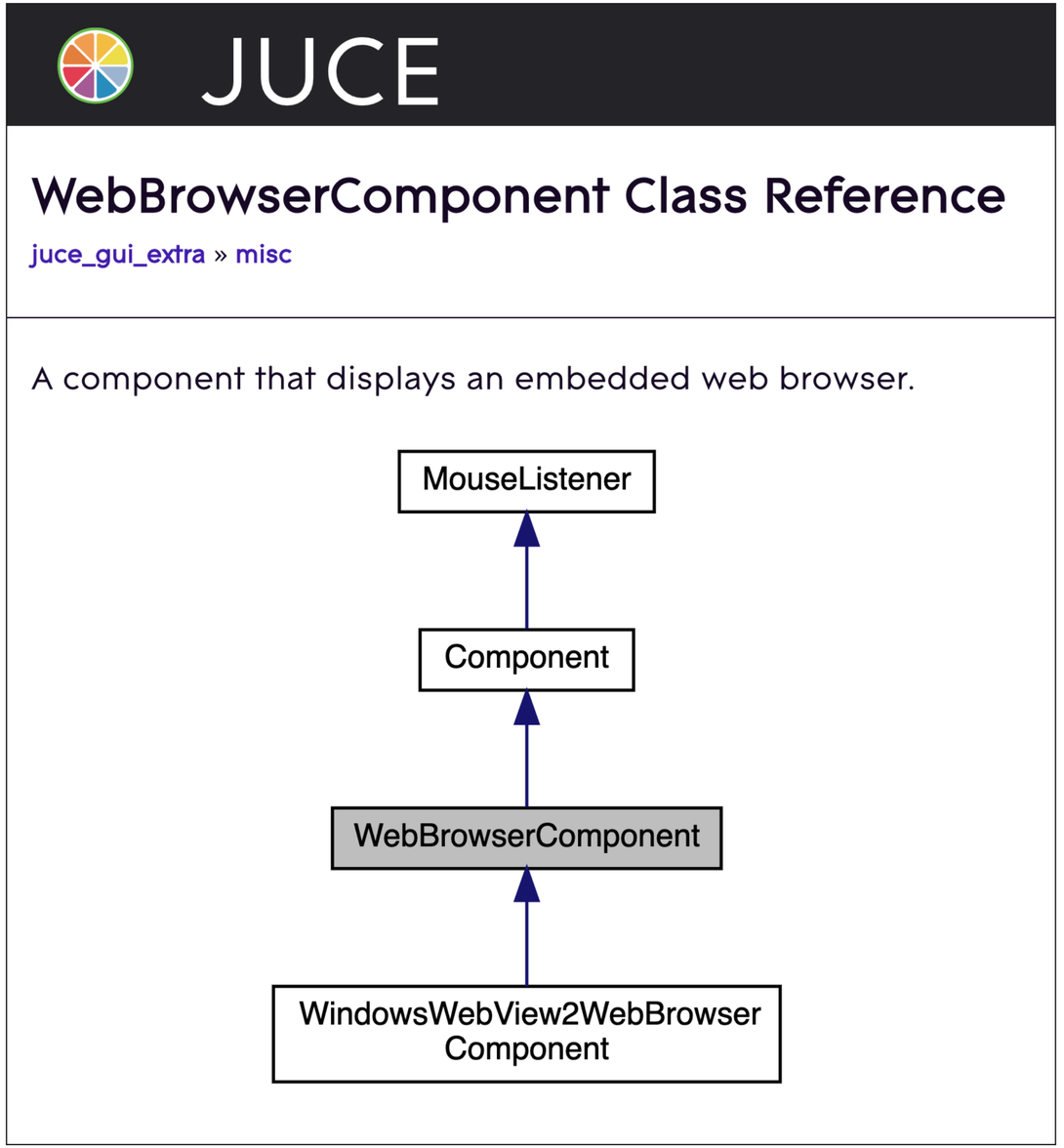


1. Prototyping
2. Why Web Tech?
3. Case Study: Arpeggiator
4. C++ / WebViews

C++ / WebViews

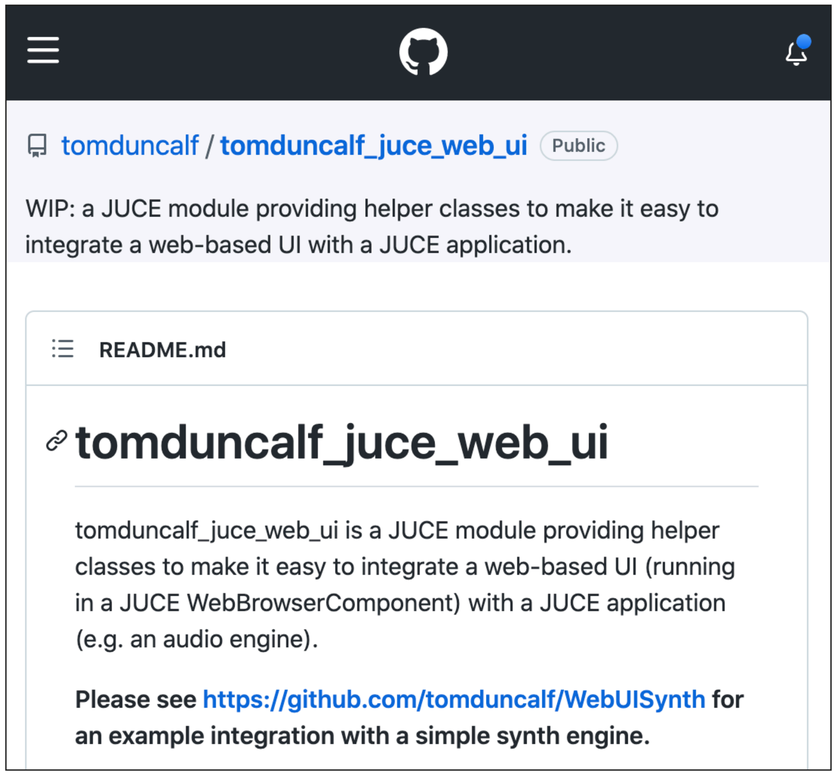

WebView
I am JUCE
ADC
Home
ADC21
Diversity
Newsletter
Sponsors
Audio Developer Conference 2021
We are delighted to invite you to attend and contribute to the 7th Audio Developer Conference (ADC)
ADC21
November 15th & 16th - In-Person/Online Conference in London --- November 19th - Online Workshops
ADC is an annual event celebrating all audio development technologies, from music applications and game audio to audio processing and embedded systems. ADC's mission is to help attendees acquire and develop new skills, and build a network that will support their career development. It is also aimed at showcasing academic research and facilitating collaborations between research and
ADC21
PURCHASE TICKETS
Purchase Tickets
ADC20
AUDIO DEVELOPER CONFERENCE
Watch ADC 2020 Talks
ADC
Sign Up for the ADC Newsletter

C++ / WebViews



https://docs.juce.com/master/classWebBrowserComponent.html

C++ ↔ JavaScript Communication



https://youtube.com/watch?v=XvsCaQd2VFE

https://github.com/tomduncalf/tomduncalf_juce_web_ui

quicktype

```
{
  "Note": {
    "type": "object",
    "properties": {
      "noteNumber":  { "type": "integer" },
      "start_s":     { "type": "number" },
      "duration_s":  { "type": "number" },
      "velocity_01": { "type": "number" }
    },
    "required": [
      "noteNumber",
      "start_s",
      "duration_s",
      "velocity_01"
    ]
  }
}
```

JSON Schema

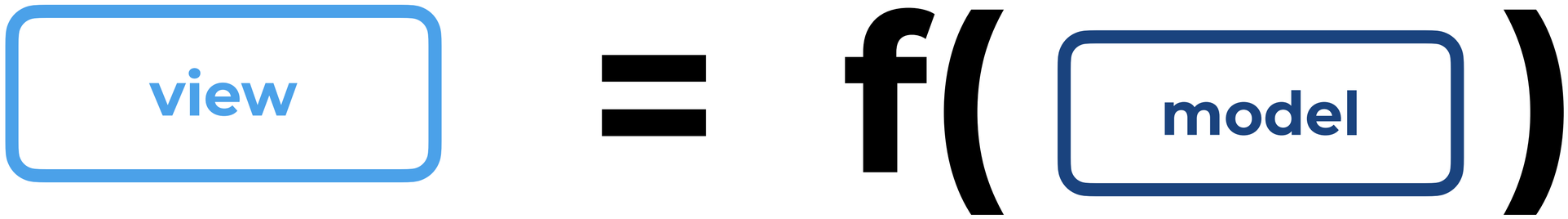
```
struct Note {
  double durationS;
  int64_t noteNumber;
  double startS;
  double velocity01;
};
```

C++

```
interface Note {
  duration_s:  number;
  noteNumber:  number;
  start_s:     number;
  velocity_01: number;
}
```

TypeScript

```
namespace nlohmann {
    void from_json(const json & j, MyApp::Note& x) {
        x.durationS = j.at("duration_s").get<double>();
        x.noteNumber = j.at("noteNumber").get<int64_t>();
        x.startS = j.at("start_s").get<double>();
        x.velocity01 = j.at("velocity_01").get<double>();
    }

    void to_json(json & j, const MyApp::Note & x) {
        j = json::object();
        j["duration_s"] = x.durationS;
        j["noteNumber"] = x.noteNumber;
        j["start_s"] = x.startS;
        j["velocity_01"] = x.velocity01;
    }
}
```


view
=
f(
model
)

C++ / WebViews

```
export interface IState {
    tempo: number;
    activeLayer: number;
    inputNotes: { noteNumber: number; velocity_n: number };
    arpType: "Up" | "Down";
    layers: Array<{
        id: string;
        name: string;
        isEnabled: boolean;
        adders: {
            volume_n: number;
            length_n: number;
            octave: number;
            chordProbability_n: number;
            panning: number;
            chance_n: number;
        };
        sequence: Array<{
            rate: number;
            numActiveSteps: number;
            steps: Array<{
                volume_n: number;
                length_n: number;
                octave: number;
                chordProbability_n: number;
                panning: number;
                chance_n: number;
            }>;
        }>;
        activeStepProperty: "volume_n" | "length_n" | "octave" | "chordProbability_n" | "panning" | "chance_n";
        speed: number;
        reps: number;
        timeSig: {
            numerator: number;
            denominator: number;
        };
        swing: number;
        randomMode: "current" | "all";
    }>;
}
```

Unidirectional data flow (simplified)

model

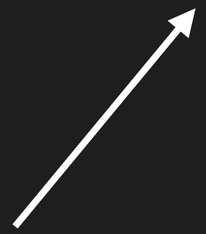
view

render(state);

update(newState);

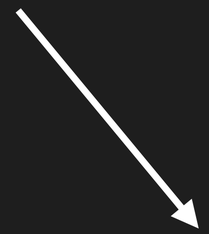
sendMessage(type, data);

localStorage



save

load



```
localStorage.setItem("viewState",
                     JSON.stringify(viewState));
```

```
JSON.parse(localStorage.getItem("viewState"));
```

onViewStateChanged

onPageLoad

thi.ng/umbrella

typescript datastructure 2d clojure 3d workshop array binary geometry browser animation clojurescript codegen graph c canvas functional processing.org visualization toxiclibs ui webgl export fileformat graphics math svg conversion dsl hiccup literate-programming tree audio interval typedarray uk acceleration async bezier color generative hardware iterator java networked parser transducer cli clipping design image random bbox circle fabrication interpolation mesh simulation baremetal component glsl nd particles query rstream architecture dsp generator polygon shader stm32 stream analysis ast composition database gui physics reactive sample sort vector agent align declarative format gpgpu html line matrix voxel wasm adjacency arc channel dataflow distance filter gradient memory-mapped points shape subdivision typography alpha aos benchmark chart css diff event logger polyline rgb string 16bit 1d 8bit ansi area bigint blend compiler data-oriented dom encode fuzzy grid hdom immutable intersection noise opencl opengl oscillator pipeline pixel rect render semweb serbia server text topology triples 24bit bitwise cache cartesian cellular-automata circumcenter concatenative cubic dag fft forth germany graphviz hash language logic macro node object paper quaternion recursive scenegraph sculpture sensor state theme union xml 32bit allocator ascii assembly bitmap blit branding convex curve datalog decode dependency drawing dual equality

In conclusion...

Do prototyping

Value iteration time

Slides and references

arthurcarabott.com/adc-2021

Contact (we are hiring)

arthur.carabott@output.com

 @acarabott