Advanced Coding Tools and Methodologies

Prof. **Francesco Bruschi**

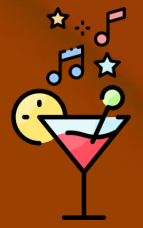
Prof. **Vincenzo Rana**

Music and Acoustic Engineering M.Sc.

Student

Marco Muraro

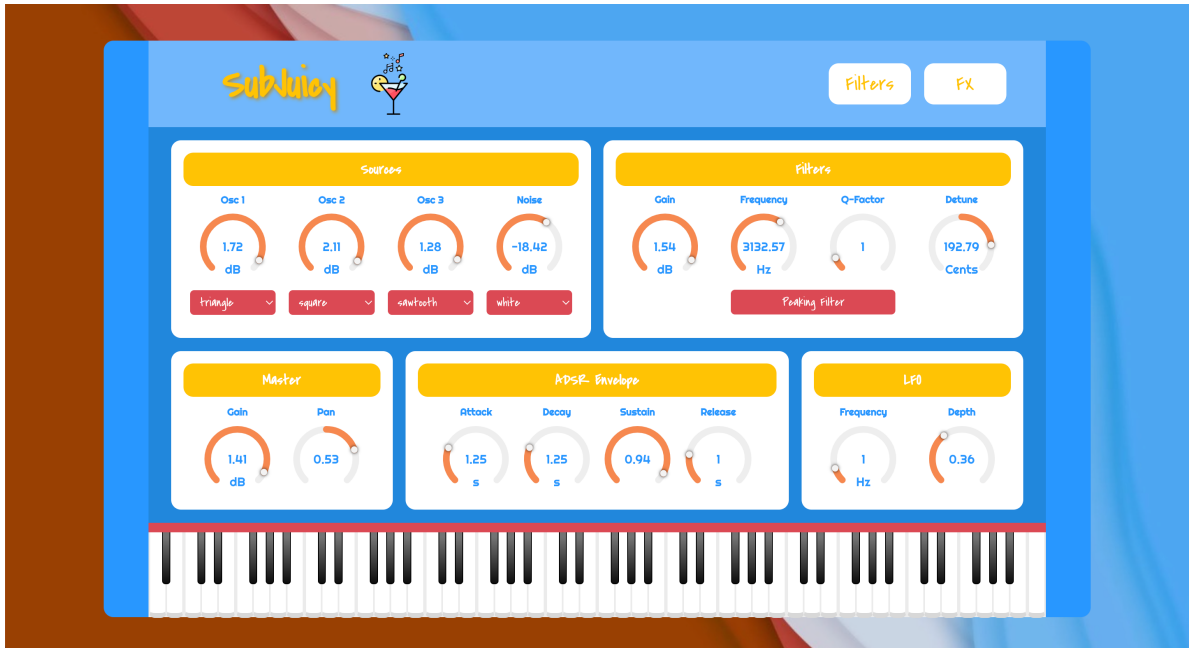
SubJuicy



Subtractive Synthesizer

POLITECNICO MILANO 1863

MAE MASTER OF SCIENCE MUSIC AND ACOUSTIC ENGINEERING



SubJuicy

Vue3 Web App

Monophonic synthesizer based on subtractive synthesis

Libraries employed

- **Vue.js**
- **Tone.js**

Web MIDI API is exploited to manage MIDI device connection and messages

POLITECNICO
MILANO 1863

Vue.js

Progressive JavaScript Framework for building web user interfaces

Approchable

Builds on top of standard HTML, CSS and JavaScript

Performant

Truly, reactive compiler-optimized rendering system

Versatile

Rich, incrementally adaptable ecosystem

Vue CLI Standard tooling baseline for Vue ecosystem

Tone.js

Web Audio Framework for creating interactive music in the browser

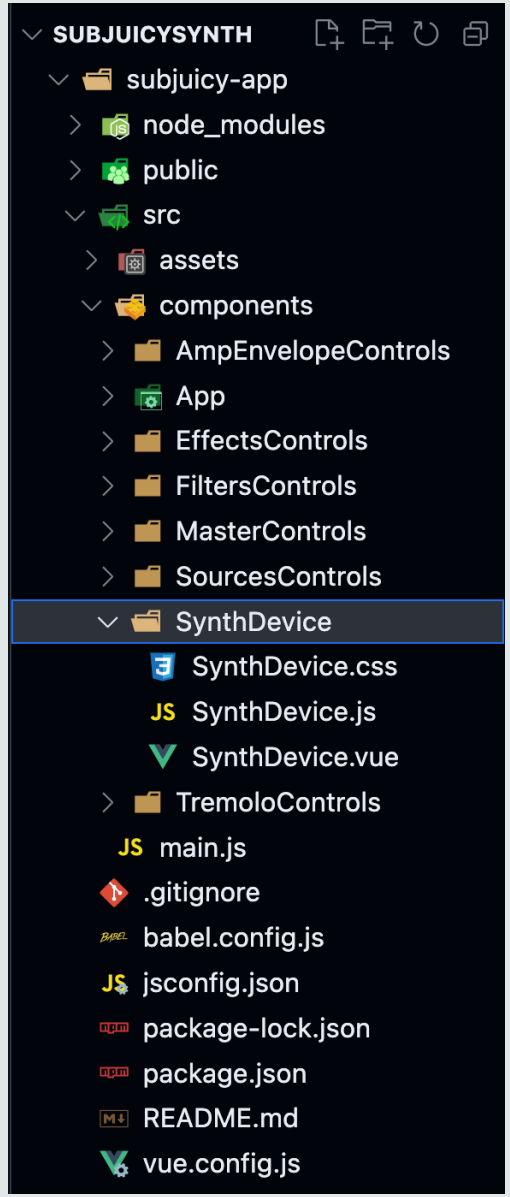
On the **high level**, Tone.js offers

- Common DAW features like global transport as well as interesting prebuilt synths and effects of different nature
- High-performance building blocks to create your own synthesizers, effects and complex control signals

App Architecture

Functional macro-sections integrated within the GUI of SubJuicy app are implemented as **Vue components**

- **Independent and reusable blocks** that are integrated in the app
- Vue **apps** are commonly **organized into nested components**
- Vue implements its own **component model** that allows us to encapsulate **custom content and logic** in each one of them



Single Component

MasterControls.vue | MasterControls.js | MasterControls.css

subjuicy-app > src > components > MasterControls > MasterControls.vue > { } style scoped

```vue
<template>
    <div class="component-wrapper flex-container-col">
        <div class="title-container">
            <h2>Master</h2>
        </div>
        <div class="flex-container-row controls-wrapper">…
        </div>
    </div>
</template>

<script src="./MasterControls.js"></script>

<style src="./MasterControls.css" scoped></style>
```

MasterControls.vue | MasterControls.js | MasterControls.css

subjuicy-app > src > components > MasterControls > JS MasterControls.js > default

```js
import RoundSlider from 'vue-three-round-slider'

export default {
    name: 'MasterControls',
    components: {
        RoundSlider
    },
    props: {
        synthState: Object
    },
    data() {
        return {
            roundSlider: {…
            },
            masterGain: this.synthState.master.gain,
            pan: this.synthState.master.pan
        }
    },
    watch: {
        masterGain() {…
        },
        pan() {…
        }
    }
}
```

MasterControls.vue | MasterControls.js | MasterControls.css

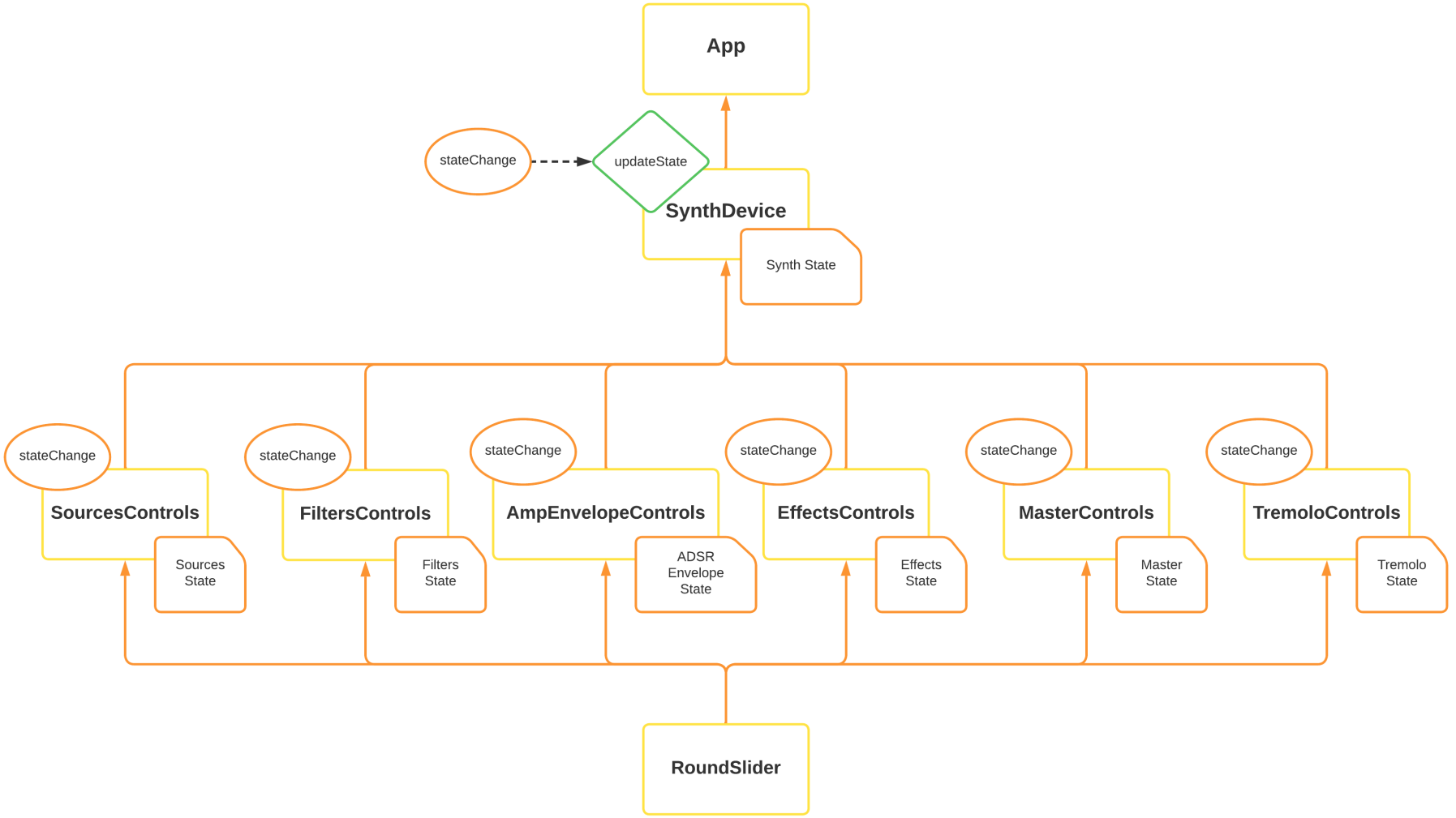
subjuicy-app > src > components > MasterControls > MasterControls.css > ...

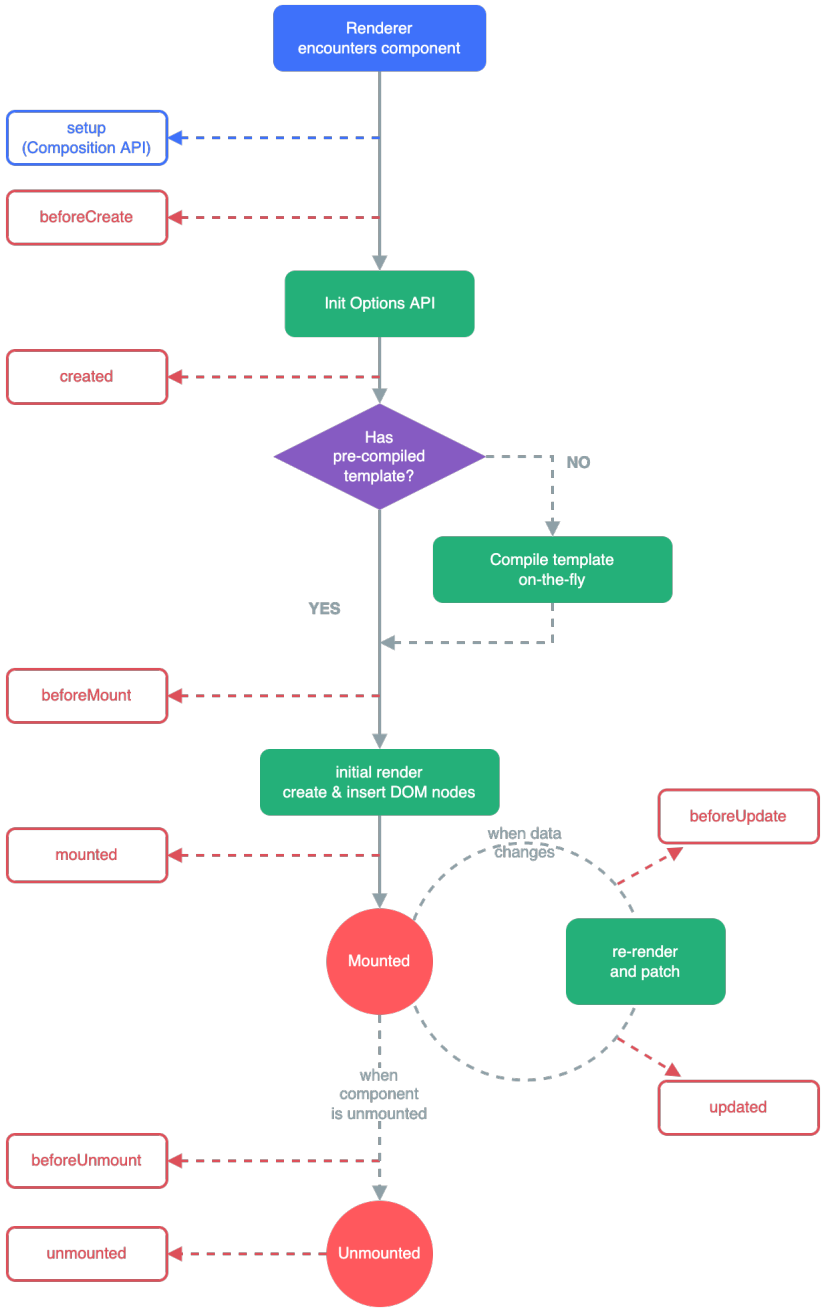
```css
.component-wrapper {
    box-sizing: border-box;
    width: 100%;
    height: 100%;
    border-radius: 12px;
    padding: 15px;
}

.title-container {
    width: 100%;
    border-radius: 12px;
    background-color: #ffc304;
}

.controls-wrapper {…
}

.knob-wrapper {…
}

label {…
}

.unit-of-measure {…
}
```

POLITECNICO MILANO 1863

POLITECNICO MILANO 1863

App Architecture





Lifecycle Hooks

Each Vue component goes through a series of initialization steps when it's created and then mounted to the DOM

Along the way, Vue runs functions called **lifecycle hooks**

Developers can add their own code to be executed at specific stages during lifecycle of a certain Vue instance

SynthDevice.vue | SynthDevice.js | SynthDevice.css

subjuicy-app > src > components > SynthDevice > SynthDevice.js > default > created

```
404     created() {
405
406         /* ----- CREATE and INITIALIZE AUDIO NODES ----- */
407
408         // Sources
409         this.osc1 = new this.$tone.Oscillator({
410             'type': this.synthState.sources.osc1.type,
411             'volume': this.synthState.sources.osc1.volume
412         });
```

```
synthState: {
    sources: {
        osc1: {
            type: 'sine',
            volume: 0
        },
        osc2: {…
        },
        osc3: {…
        },
        noise: {
            type: 'white',
            volume: -50
        }
    },
    amplitudeEnvelope: {…
    },
    master: {…
    },
    lfo: {…
    },
    filters: {…
    },
    effects: {…
    }
}
```

Synth State

Synth state is defined as an object in the **SynthDevice component** within its data function

SynthDevice is the **heart of signal processing** for the whole app

synthState object is exploited to initialize all the audio nodes and is also passed as a property to all the child components for GUI controls initialization

```
this.peakingFilter = new this.$tone.BiquadFilter({
    'type': 'peaking',
    'gain': this.synthState.filters.peaking.gain,
    'frequency': this.synthState.filters.lowPass.frequency,
    'Q': this.synthState.filters.lowPass.quality,
    'detune': this.synthState.filters.lowPass.detune
});
```

State Update

SourcesControls.vue | SourcesControls.js | SourcesControls.css | SynthDevice.js

subjuicy-app > src > components > SourcesControls > SourcesControls.js > ...

```
38      watch: {
39          osc1Type() {
40              console.log('Osc1 Type chnaged:', this.osc1Type);
41              this.$emit("stateChange", {name: 'osc1Type', value: this.osc1Type});
42          },
43          osc1Volume() {
44              console.log('Osc1 Volume changed:', this.osc1Volume);
45              this.$emit("stateChange", {name: 'osc1Volume', value: this.osc1Volume});
46          },
```

SynthDevice.vue | SynthDevice.js | SynthDevice.css | SourcesControls.js

SynthDevice.vue > { } template > div#synth.flex-container-row > div#synth-panel.flex-container-col

```
22                  <div id="controls-panel" class="flex-container-col">
23                      <div class="controls-wrapper flex-container-row">
24                          <div id="sources-controls">
25                              <SourcesControls
26                              :synthState="synthState"
27                              @stateChange="updateState"/>
28                          </div>
```

SynthDevice.vue | SynthDevice.js

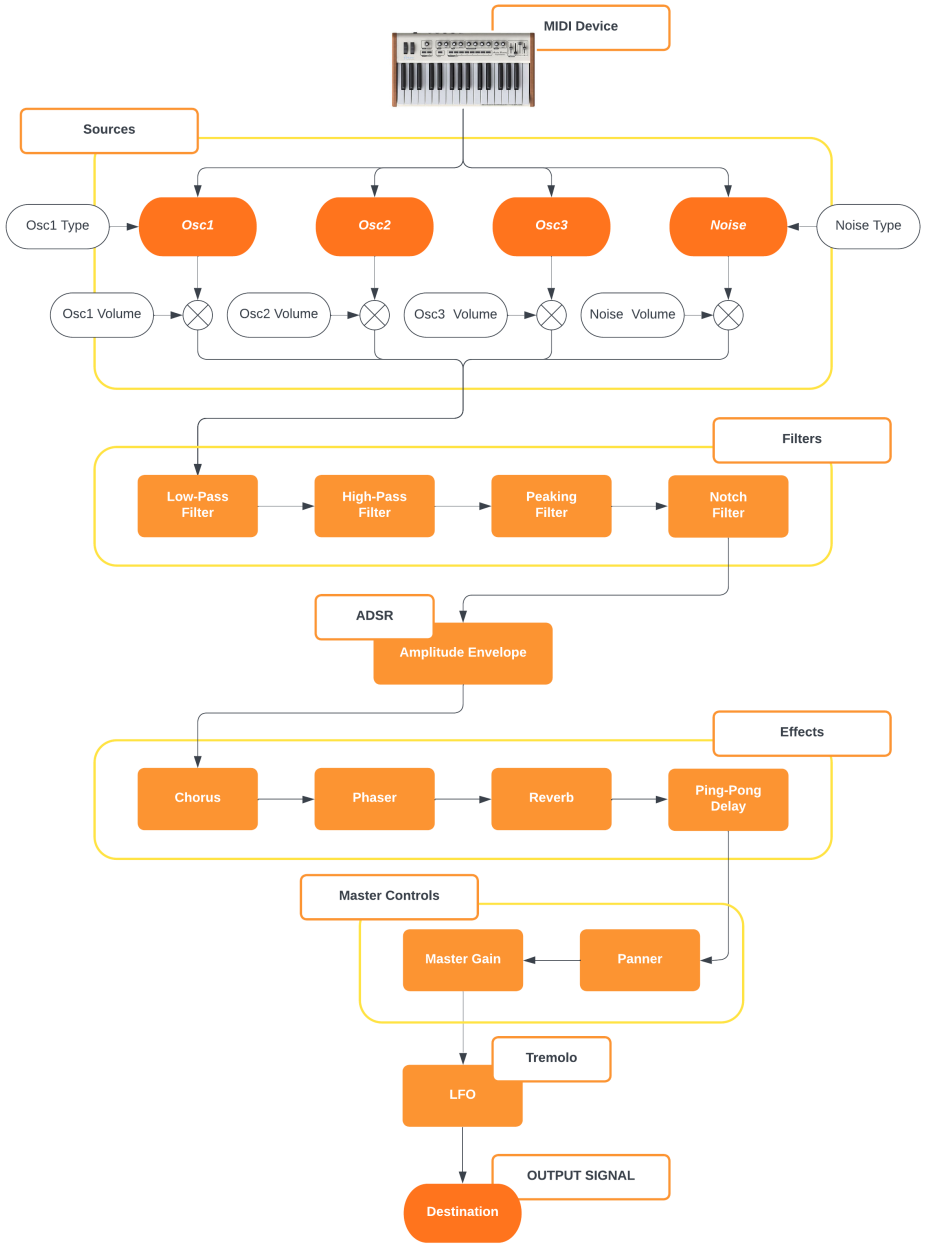
subjuicy-app > src > components > SynthDevice

```
184         updateState(update) {
185             switch(update.name) {
186                 case 'osc1Type':
187                     this.synthState.sources.osc1.type = update.value;
188                     this.osc1.type = this.synthState.sources.osc1.type;
189                     console.log('Osc1 Type updated:', update.value);
190                     break;
191                 case 'osc1Volume':
192                     this.synthState.sources.osc1.volume = update.value;
193                     this.osc1.volume.value = this.synthState.sources.osc1.volume;
194                     console.log('Osc1 Volume updated:', update.value);
195                     break;
```

v-model directive automatically updates child component state

Vue watchers are employed to *emit stateChange custom event* that is dispatched to the parent SynthDevice whenever the user modifies any synth parameter

POLITECNICO
MILANO 1863



POLITECNICO MILANO 1863

Signal Flow

Signal processing is implemented through **Tone.js audio nodes** that are connected together to build the overall signal flow architecture


SubJuicy
Filters
FX
Sources
Osc 1
1.88
dB
Osc 2
1.58
dB
Osc 3
2.12
dB
Noise
-18.82
dB
sine
triangle
square
pink
Effects
Depth
0.74
Delay Time
5.44
ms
Frequency
1.5
Hz
Feedback
0.34
Wet
0.36
Chorus
Master
Gain
1.58
dB
Pan
0.74
ADSR Envelope
Attack
1.29
s
Decay
1.54
s
Sustain
0.91
Release
1
s
LFO
Frequency
1
Hz
Depth
0.39

Let's jump to

SubJuicy

...and make some music

https://marcomuraro4.github.io/SubJuicy/

Advanced Coding Tools and Methodologies

Prof. **Francesco Bruschi**
Prof. **Vincenzo Rana**

SubJuicy

Thank You

Music and Acoustic Engineering M.Sc.

Student
Marco Muraro

POLITECNICO MILANO 1863

MAE MASTER OF SCIENCE MUSIC AND ACOUSTIC ENGINEERING