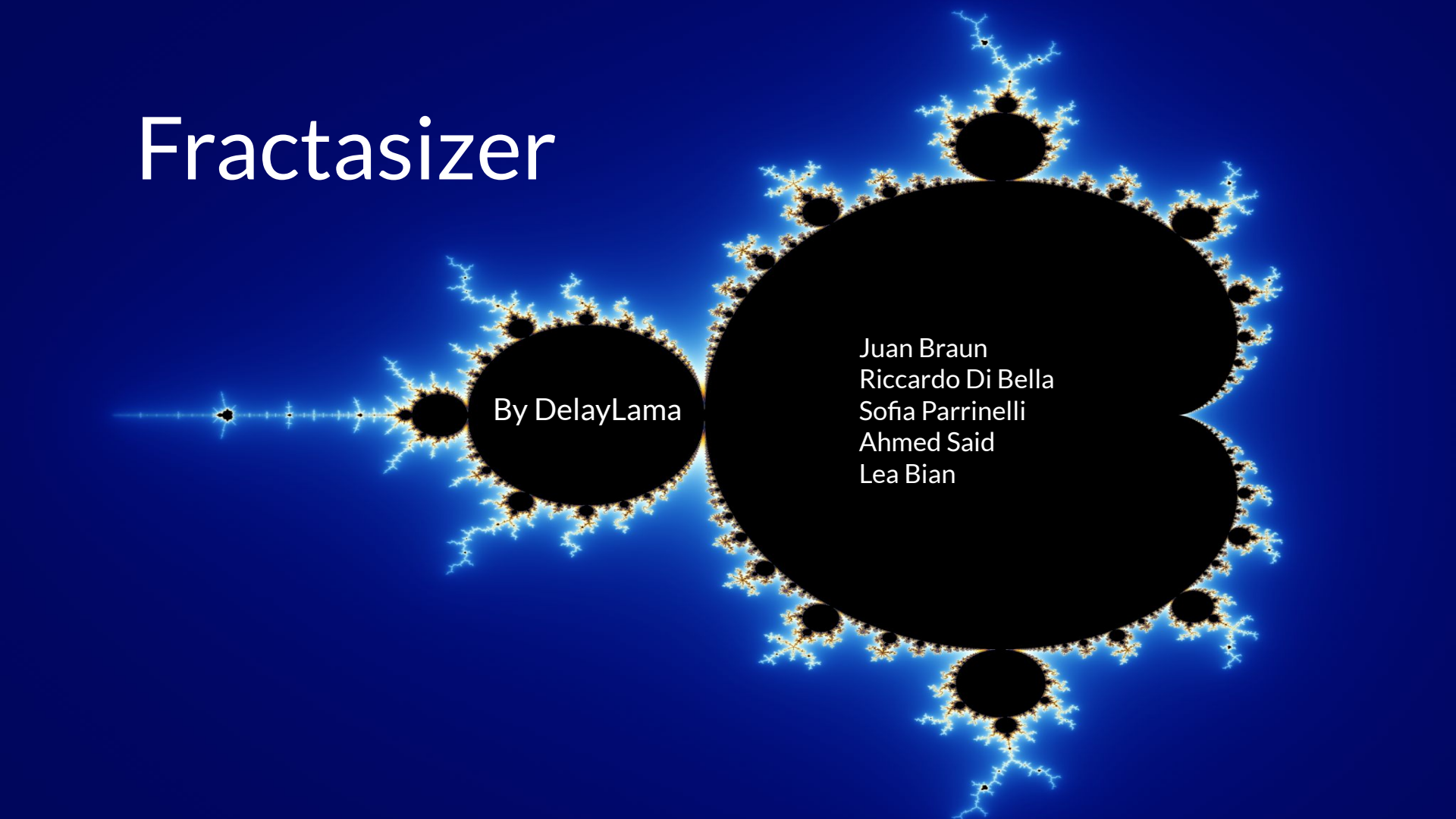
Fractasizer



By DelayLama

Juan Braun
Riccardo Di Bella
Sofia Parrinelli
Ahmed Said
Lea Bian

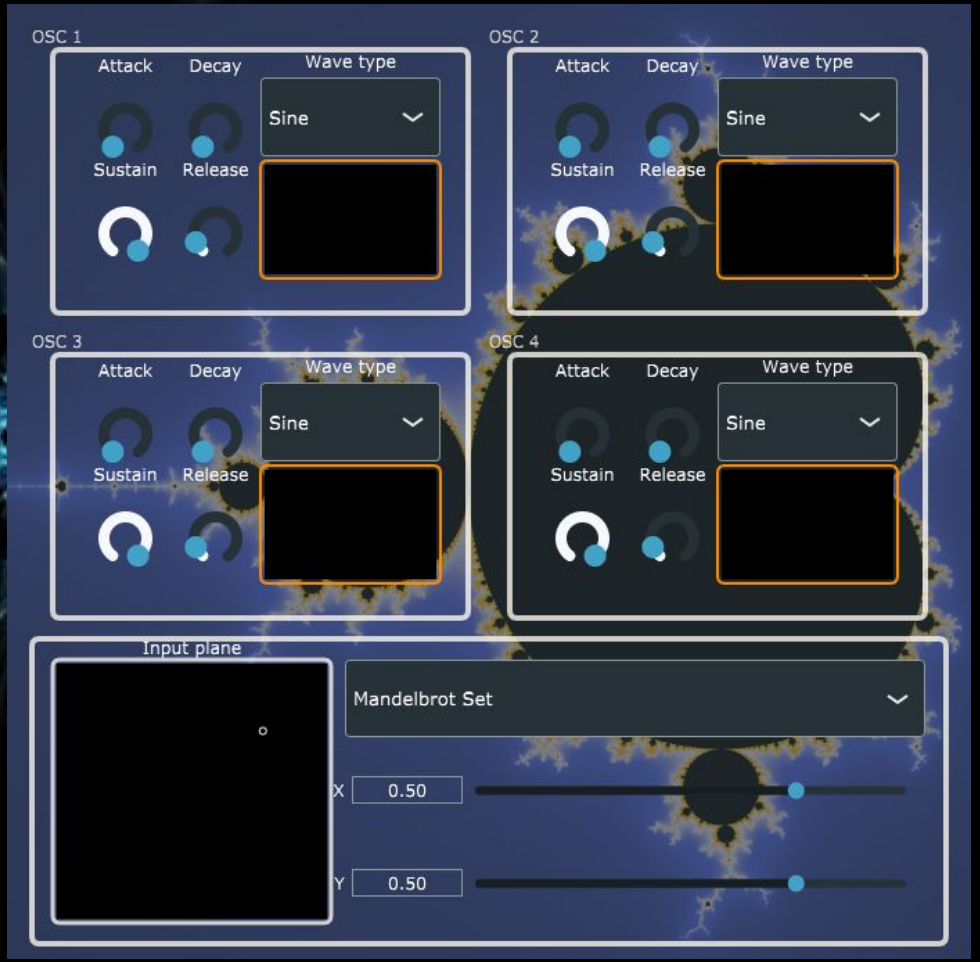
Overview

- MIDI synthesizer based on fractal systems using C++ and the JUCE framework.
- The final sound is generated by additive synthesis of four partials.
- Use of escape-time fractal formulas to compute values in order to control the parameters of an oscillator.
- Up to 10 polyphonies.

Features and GUI

- Final output is composed of the sum of four partials

- Two main areas in the GUI
  - Partials control area
  - Fractals control area



Partials control area

one panel for each sound component

Waveform combo-box:

- Sine
- Saw
- Square

ADSR

Single wave visualization

Fractal controls area:

Input plane and sliders to select the starting point for the fractal computation

Combo-box to choose the fractal

Fractals and related sound parameters

- 4 computed points for 4 partials
  - Real part:

    used as detuning factor for the partials, by multiplying it by the fundamental frequency. (The frequency of the first partial is left unchanged)

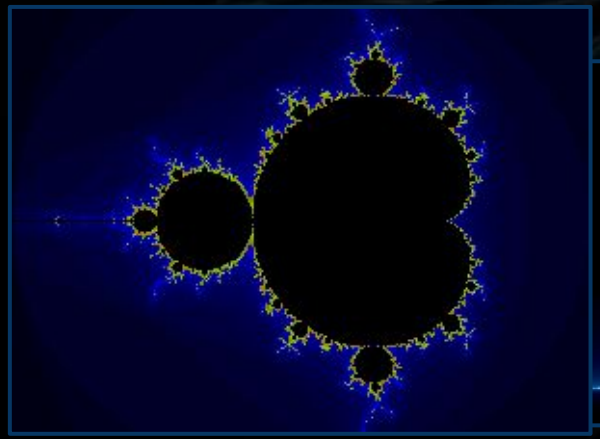
  - Imaginary part:

    used to control the rate of an LFO which controls a tremolo effect on the partials
- Fractals available
  - Mandelbrot
  - Burning Ship
  - Tricorn

Fractals

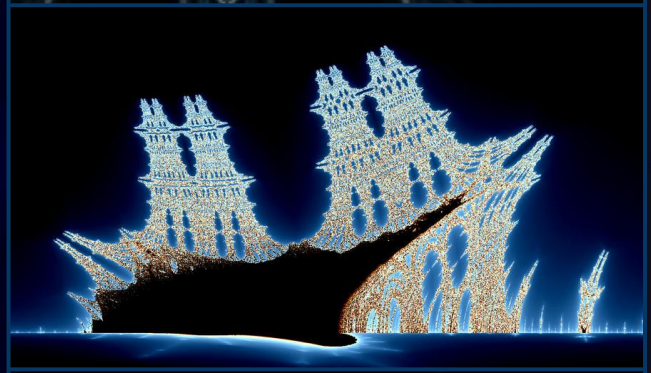
Mandelbrot set

$$z_{n+1} = z_n^{\,2} + c$$



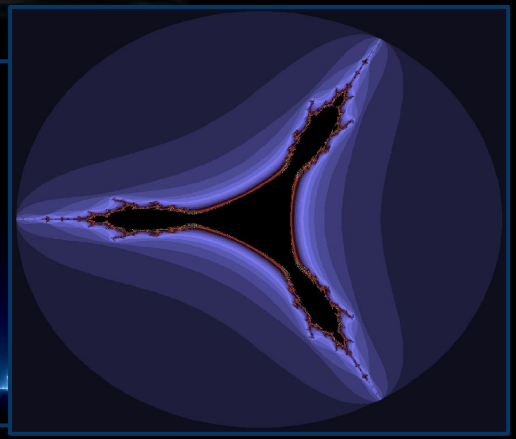
Burning ship

$$Z_{n+1} = (|\Re(Z_n)| + i|\Im(Z_n)|)^2 + c,$$



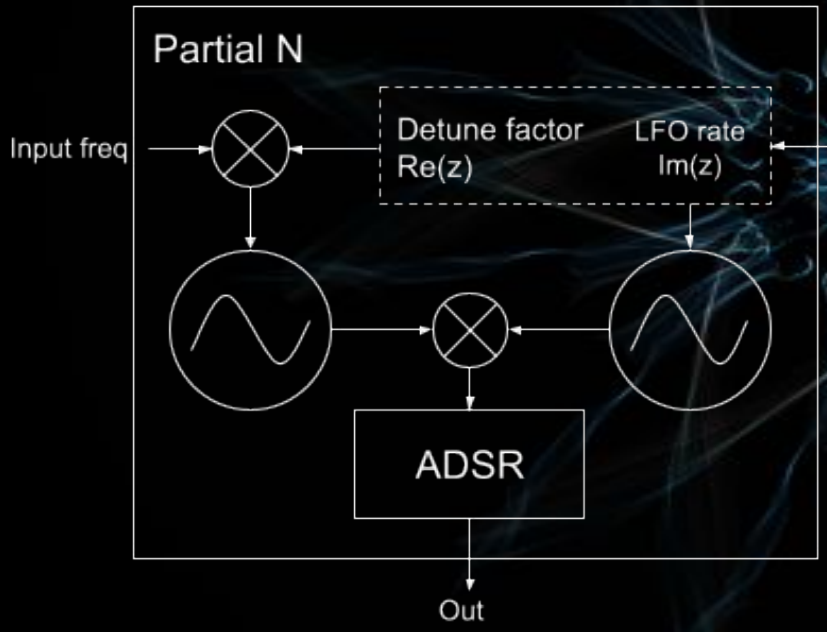
Tricorn

$$f_c : z \mapsto \bar{z}^2 + c,$$



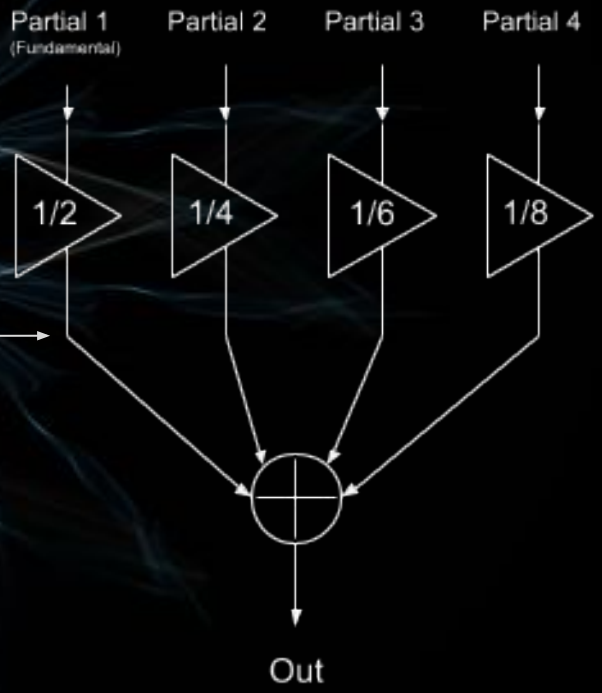
Voice generation

Diagram of the single sound component



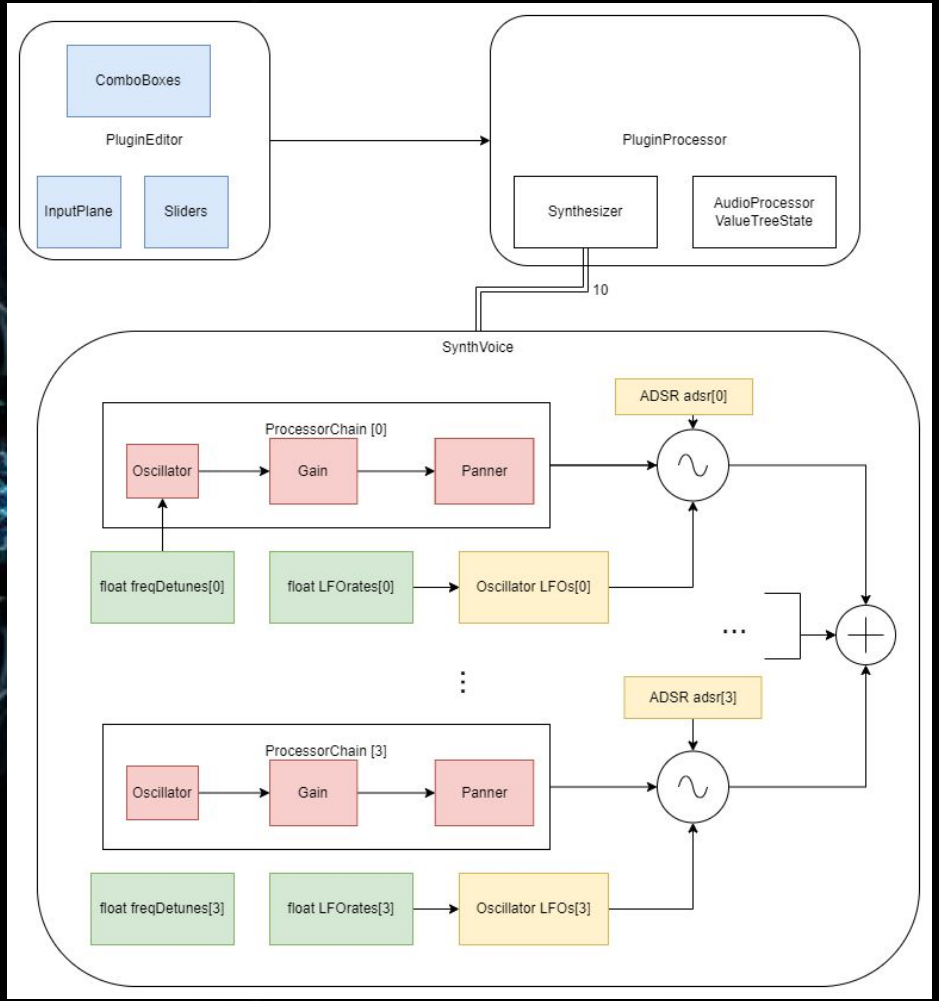
Nth succession value

Additive synthesis



Implementation details

- Based on the JUCE *Synthesiser* class, the base class to create playable MIDI instruments
- A Synthesiser contain multiples *SynthesiserVoice*
- Each *SynthVoice* contain the needed objects and variables to generate the 4 partials
- Waveforms are generated using the JUCE *Oscillator* class



Video Demo