**CSD2C Design** Harold Groenenboom

A simple polyphonic unison synthesizer with midi, filters and distortion.

*Background and motivation:* This project functions as a building block for future synthesis projects. I’ve always had a high interest for digital audio synthesis and I want to build digital (and maybe analog) synthesizers.

*Concept:* I want to make well sounding midi controlled synthesizer, which will be easy to use and must have a solid architecture. The synthesizer will have mostly standard audio synthesis components, which should make it user-friendly and easily expandable. The end result will be a stable, user-friendly, well sounding and easily expandable synthesizer.

*Learning objectives:* My aim in this project is to create a logically, easily expandable system architecture for a polyphonic midi synthesizer. In which I focus on coding the most common synthesizer building blocks which I haven’t yet worked with. The results must be esthetically pleasing at least, as the user should want to hear more of the sounds produced. The synthesizer building blocks I want to work with are:

* wavetables
* polyphony
* interpolation
* anti-aliasing
* detuning
* filters
* distortion

*Basic technical specifications:* The synthesizer will have MIDI input, the midi input will be separated into multiple voices. Every voice will have it’s own unison / multivoiced oscillator with detune containing multiple oscillator objects. These objects will generate audio by reading from a 2048 samples long wavetable and applying interpolation. The wavetables will have waves which will be generated, anti-aliased and stored inside the wavetable object. The wavetable will be send to the oscillators via a pointer. Jack2 will be used for audio generation

The MUST, COULD, and SHOULDs

MUSTs:

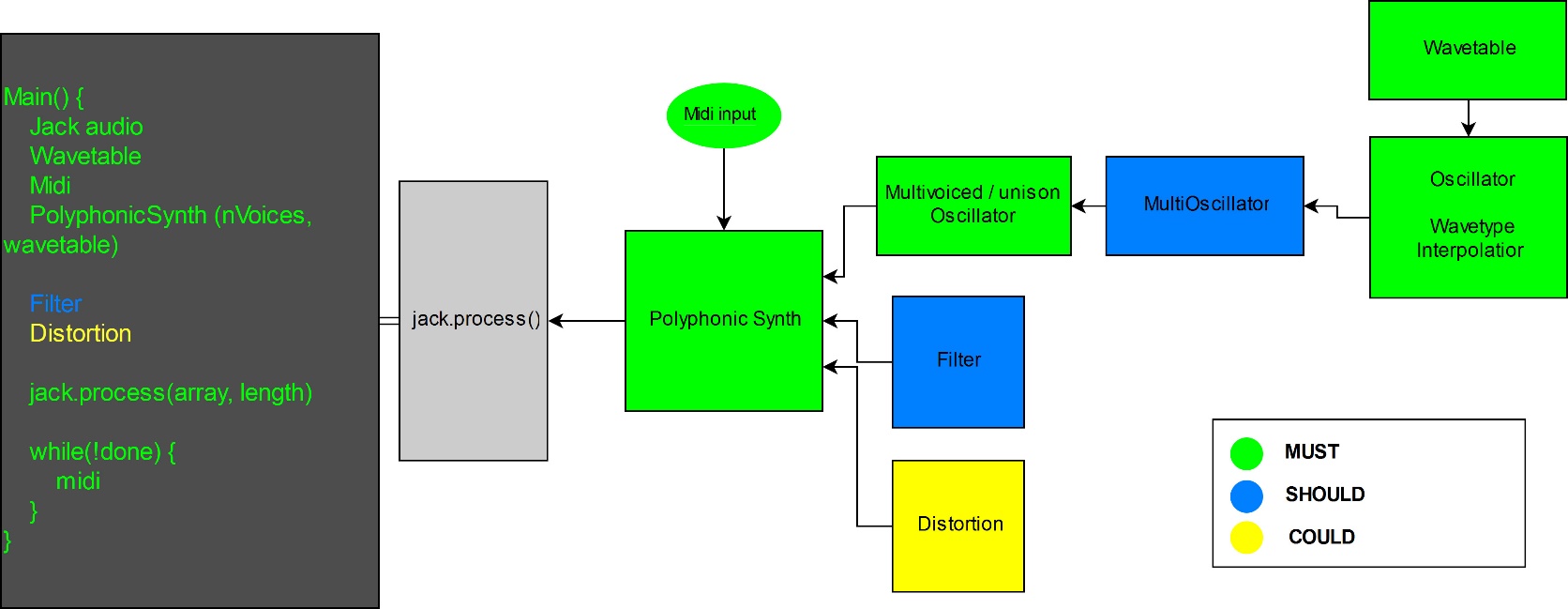
* Wavetable
  + Some kind of enumeration of single waveforms
    - i.e. vectors, audio arrays, Waveform classes or something else. The optimal solution is being researched.
  + Interpolation
  + Anti-aliasing
* Oscillator class
  + Phase, frequency and amplitude variables
  + Midi pitch to frequency conversion method
  + A specific wave selected from the wavetable
    - The interaction of the wavetable and the oscillator class is not yet fully determined. This issue is also brought up in the Wavetable research
* Polyphonic voice splitting
* Unison/multivoiced oscillator class having:
  + Number of oscillators variables
  + multiple oscillators and methods for setting the oscillators objects parameters
  + detune
* MIDI input
* Implementation of audio processing through audio callback functions.

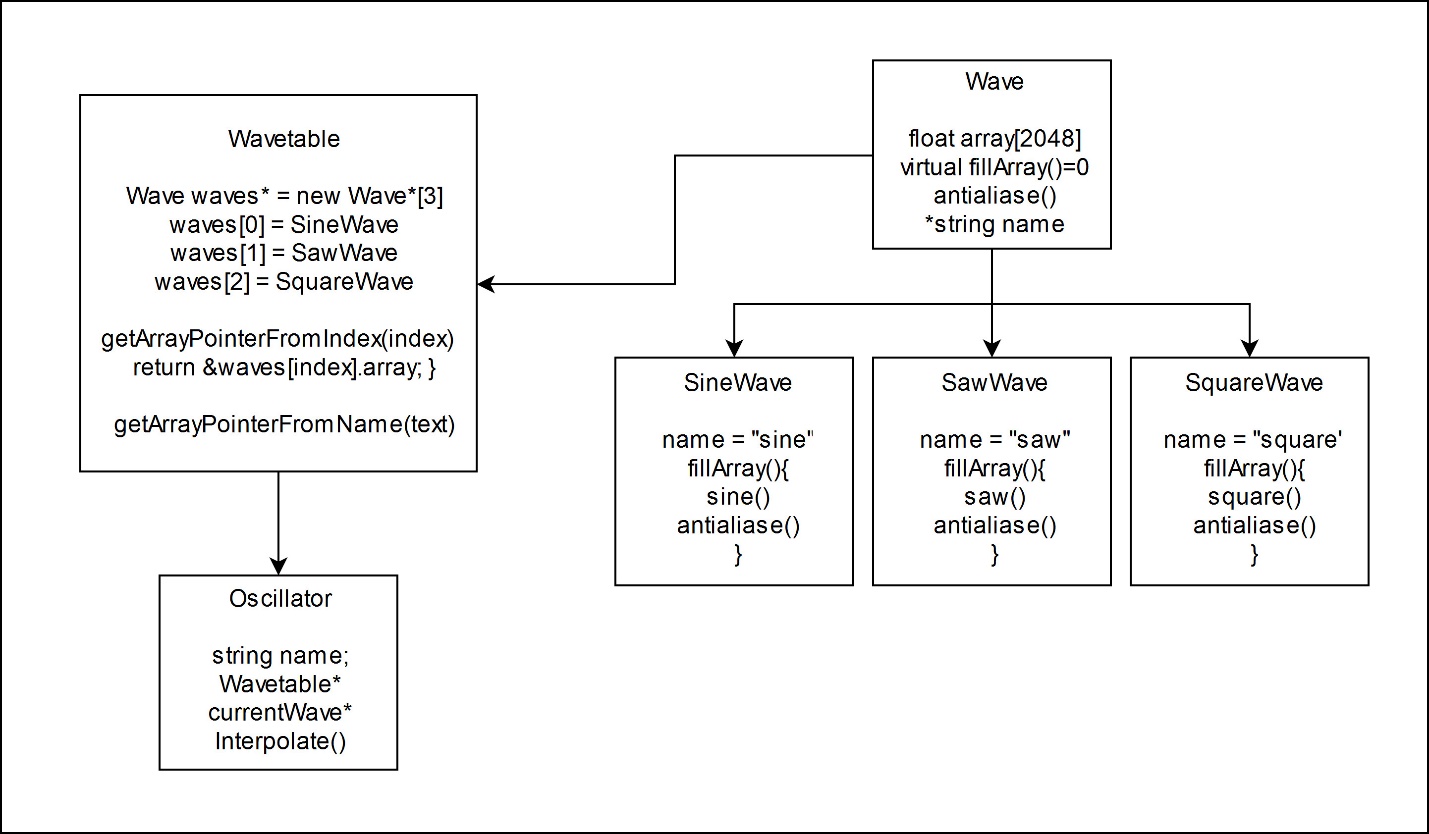
COULDs:

* Filter (lowpass), with resonance and cutoff parameters
* MultiOscillator instead of an Oscillator. The MultiOscillator has multiple oscillator objects, that together form a new more complex waveform.
  + Multiple oscillator objects
  + Pitch variable
  + Pitchoffset and amplitude variables per oscillator object.

WOULDs:

* Envelopes (ADSR)
  + Midi triggered envelopes
  + Filter & Amplitude Envelopes
* Distortion
  + Look-up table?
* LFO based morphing of the unison/multivoiced oscillator

**SYSTEM SCHEMATIC**

**WAVETABLE AND OSCILLATOR DESIGN**