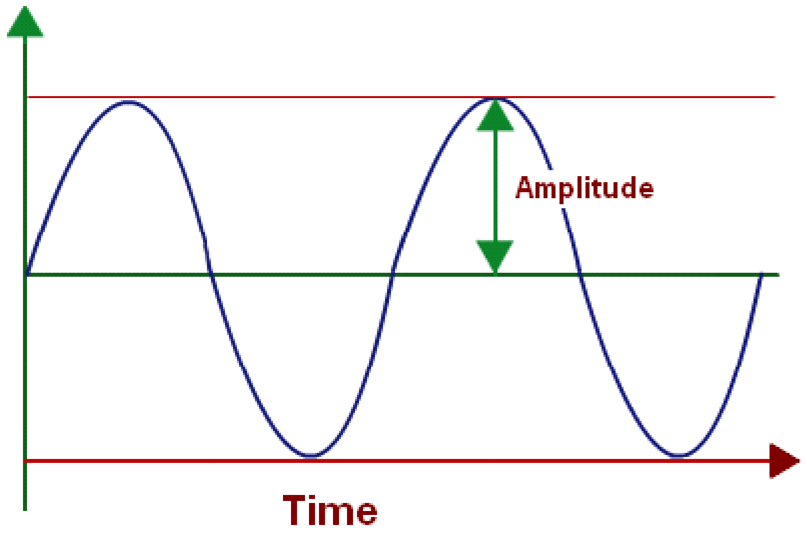
more audio with

MAXIMILIAN 0.1

Amplitude

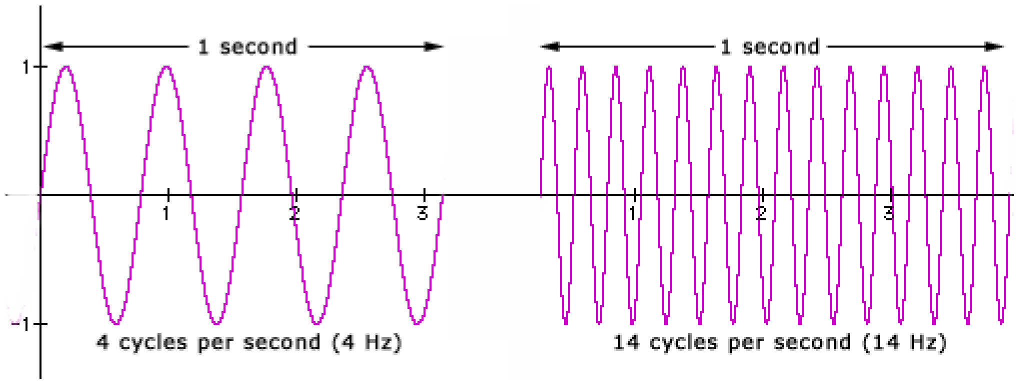
- how much does the speaker move up/down?
  - a.k.a how much air does it move?



- human perception: How loud is it?

Frequency

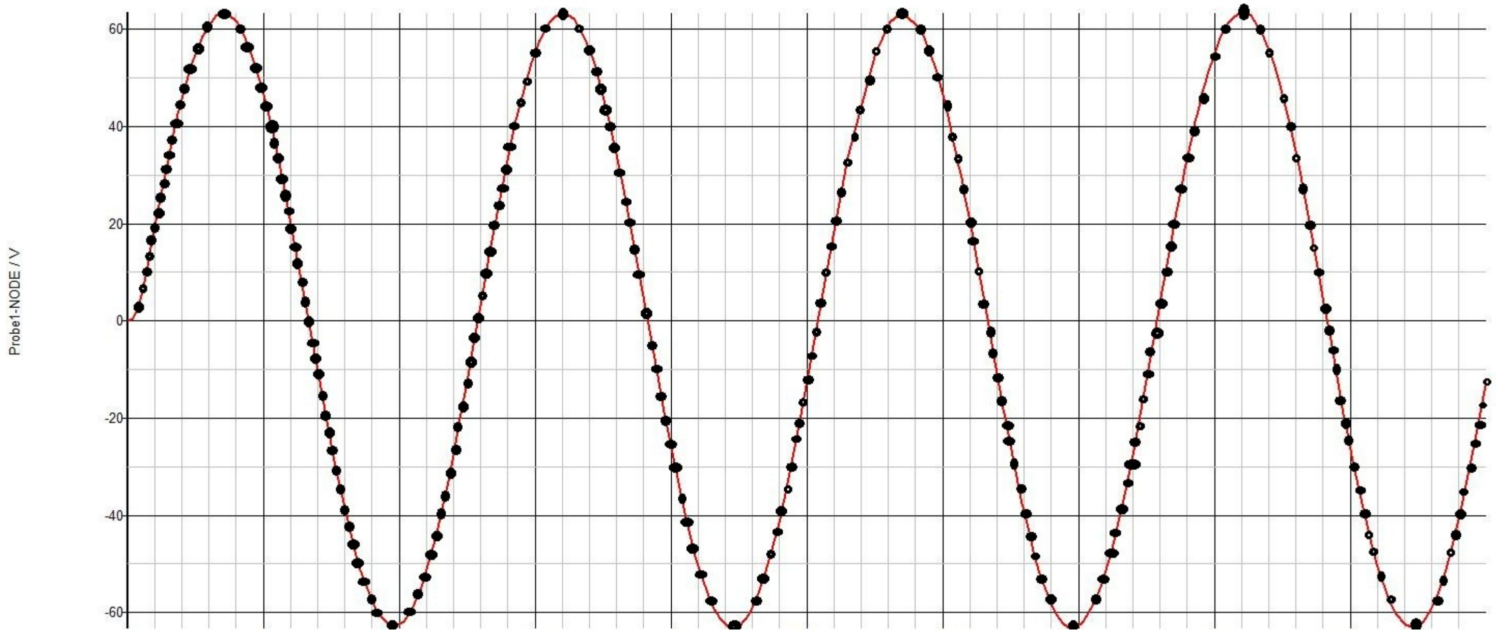
- How many times a second does a speaker go up/down?



- human perception: how pitched is the sound?

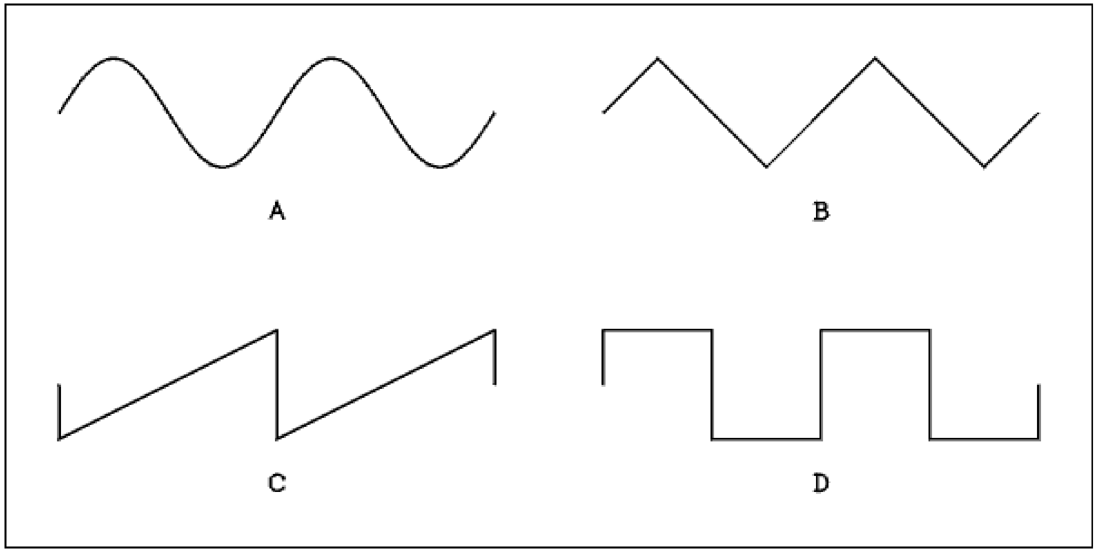
how Maximilian works

- you ask for data
- it serves it to you, one chunk at a time
- when using openFrameworks you have to fill the buffer
- the sound card calls the audioOut() function when ready
- demo: importing Maximilian examples into oFx
- bug demo!



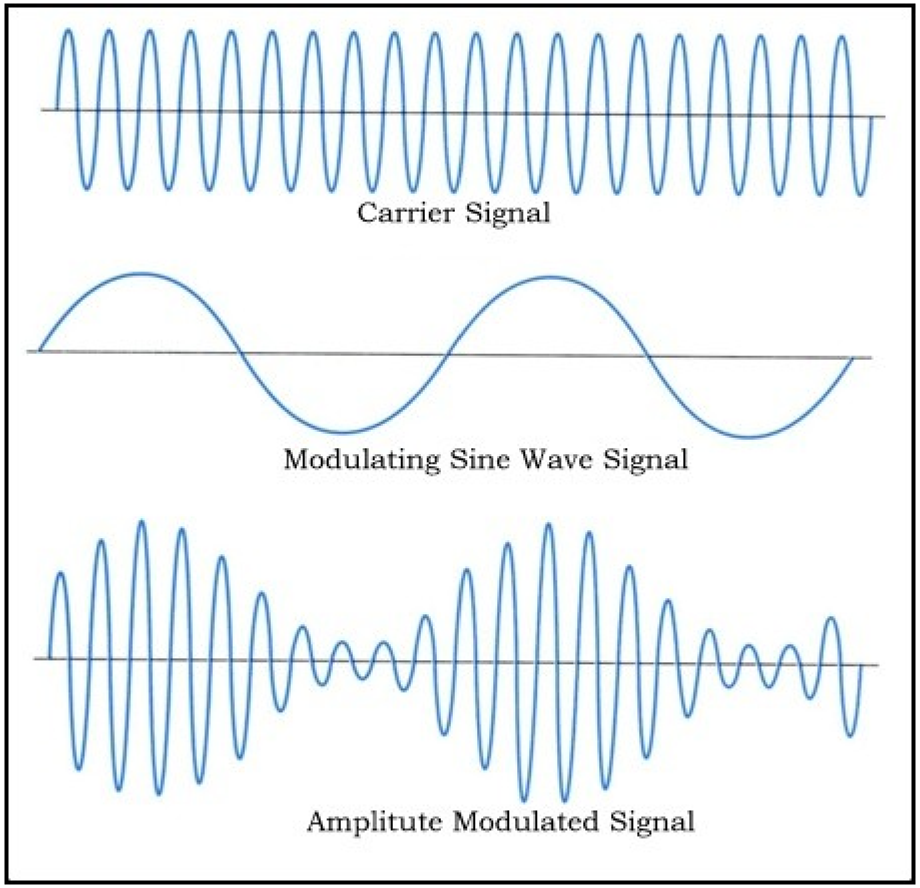
type of oscillators

- sinewave: `myOsc.sinewave(100)` `→ between -1&1, 100x/sec`
- phasor: `myOsc.phasor(0.2, 0, 300)` `→ between 0&300, 0.2x/sec (ev 5sec)`
- square: `myOsc.square(200)` `→ 1 or -1, 200x/sec`
- sawtooth: `myOsc.sawtooth(550)` `→ ???`
- see also: pulse, rect, triangle



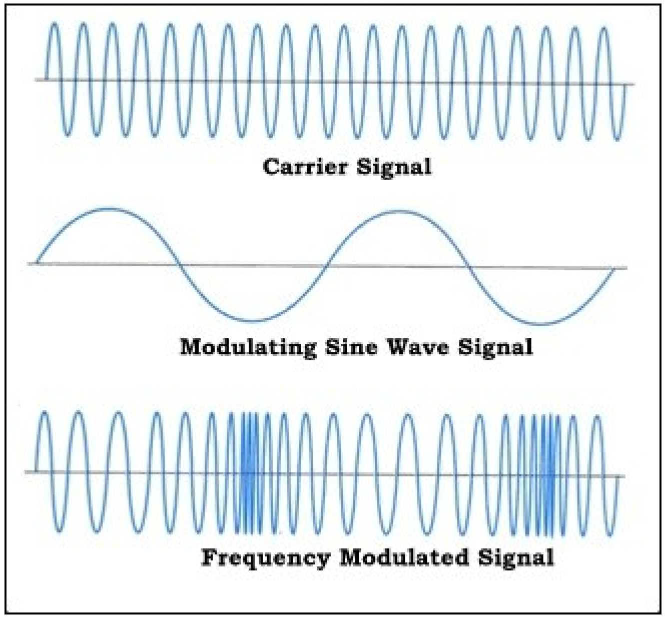
Amplitude Modulation

- carrier signal
  - "base line"
- modulation frequency
  - "how fast should it change"
  - measured in Hertz
- modulation index
  - "how much should the amplitude go up/down"



Frequency Modulation

- carrier signal
  - “base line”
- modulation frequency
  - “how fast should it change”
  - measured in Hertz
- modulation index
  - “how much should the frequency go up/down”



more combinations

```
osc1.sinewave(440) * osc2.sinewave(1)

osc1.sinewave(440) * osc2.sinewave(osc3.phasor(0.2, 1, 200))

osc1.sinewave(440 * osc2.sinewave(1))

osc1.sinewave(440 + osc2.sinewave(1) * 100)

osc1.sinewave(440 + osc2.square(1) * 100)

osc1.sinewave(440 + osc2.triangle(0.2) * 100)

osc1.sinewave(440 + osc2.sinewave(osc3.phasor(0.2,0,100)) * 100)

osc1.sinewave(440 + osc2.sinewave(osc3.sinewave(0.2) * 100)

osc1.sinewave(440 + osc2.sinewave(osc3.square(0.2)) * 100)
```

playing back sounds

- audio has to be 16 bit wav file

Sound and Visuals

how to combine them

- Feed audio data to visuals
  - create a global variable
  - audioOut() writes to it
  - draw() function reads it
- Feed visual data to audio
  - create a global variable
  - draw()/update() write to it
  - audioOut() reads it
- Use a low pass filter to get smoother results
  - what does it do?
  - shapes from agents example
  - how do we use it?

a couple more tricks

- using average / RMS
- triggering with average