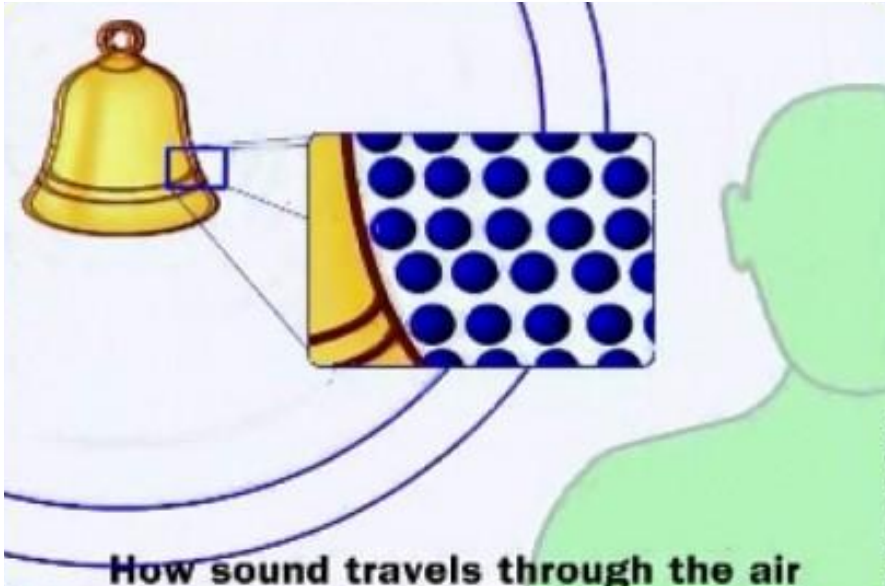
Hack Nights: Audio Synthesis

FACTLab 2015

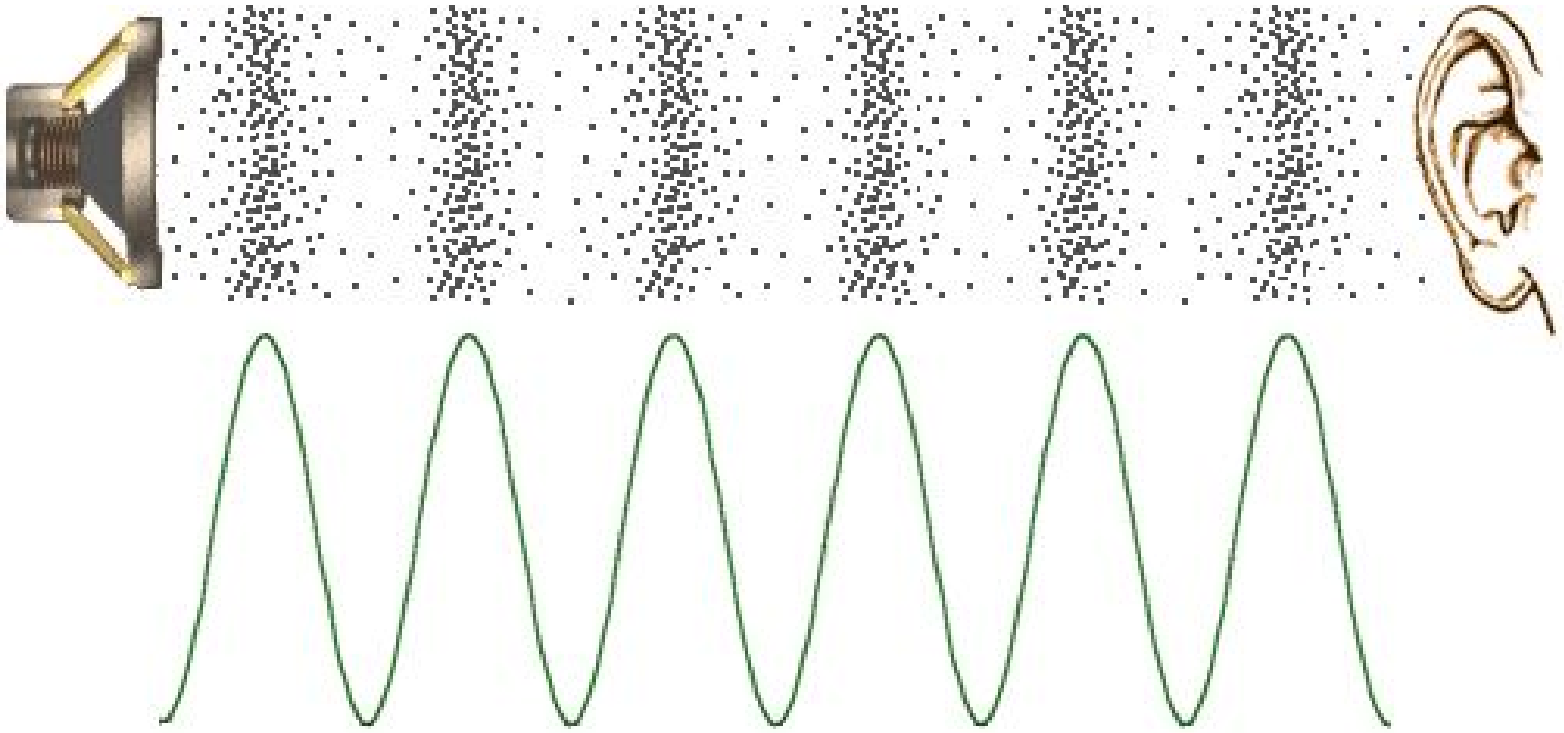
Radamés Ajna
radames.ajna@fact.co.uk

Thiago Hersan
thiago.hersan@fact.co.uk

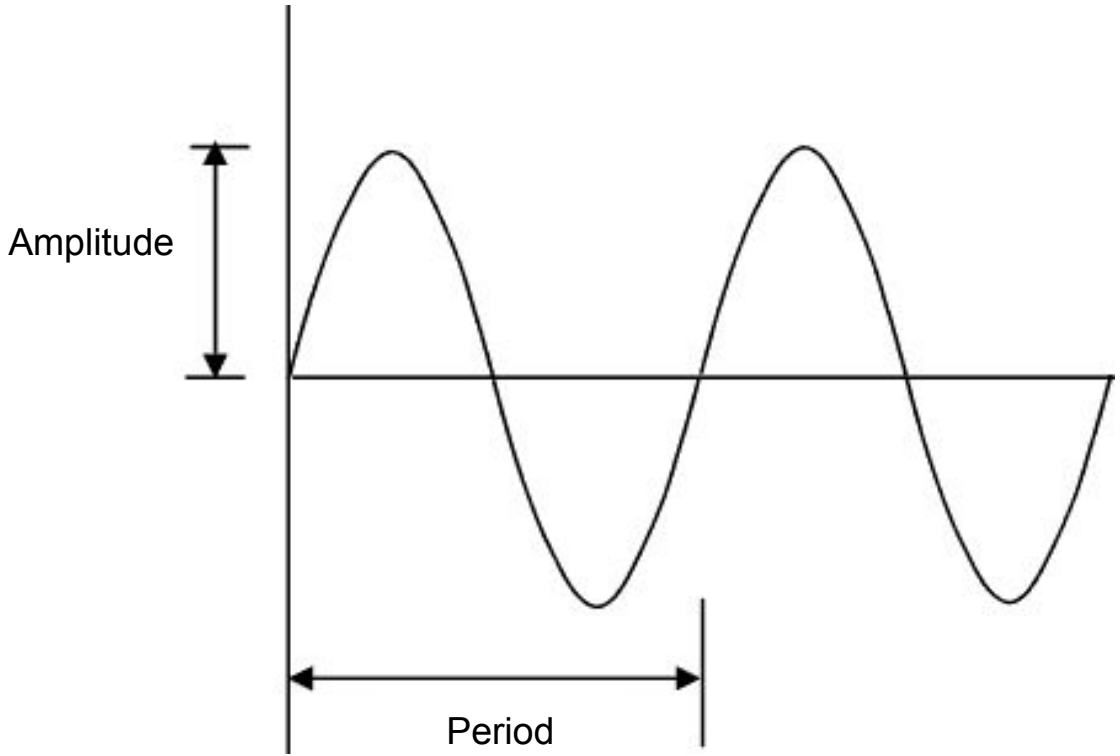
Sound and waves



Sound and waves



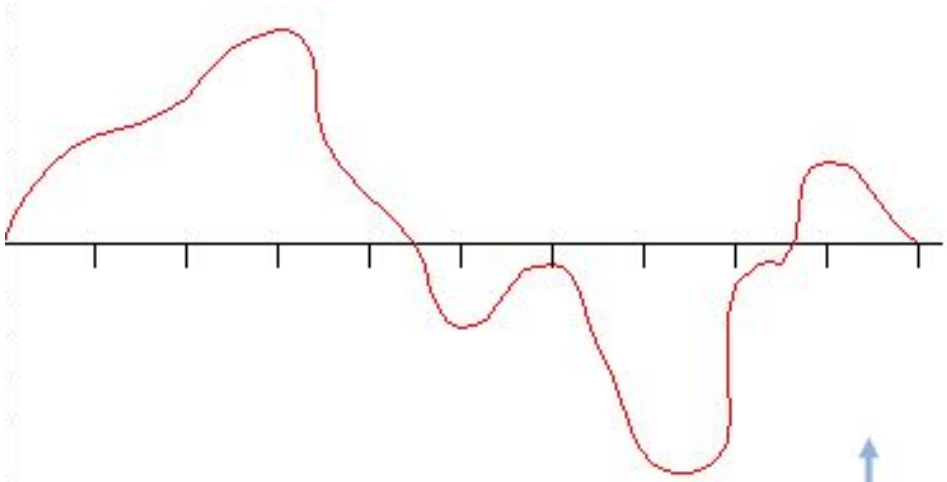
Waves

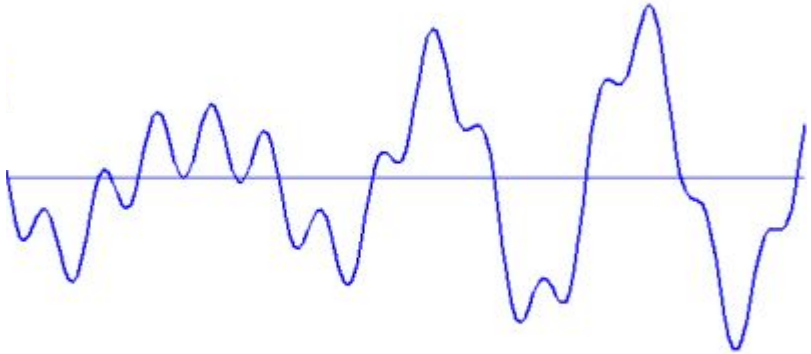


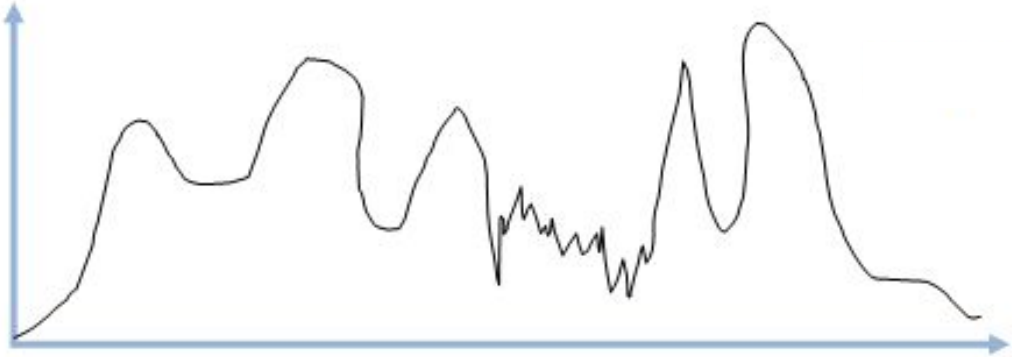
Waves in Processing

```
Minim       minim;
AudioOutput out;
Oscil       wave;

void setup() {
  size(700, 400);
  minim = new Minim(this);

  // set up an output channel
  out = minim.getLineOut(Minim.MONO, width);

  // init an oscillator and patch it to output
  wave = new Oscil(440, 0.5f, Waves.SINE);
  wave.patch(out);
}
```

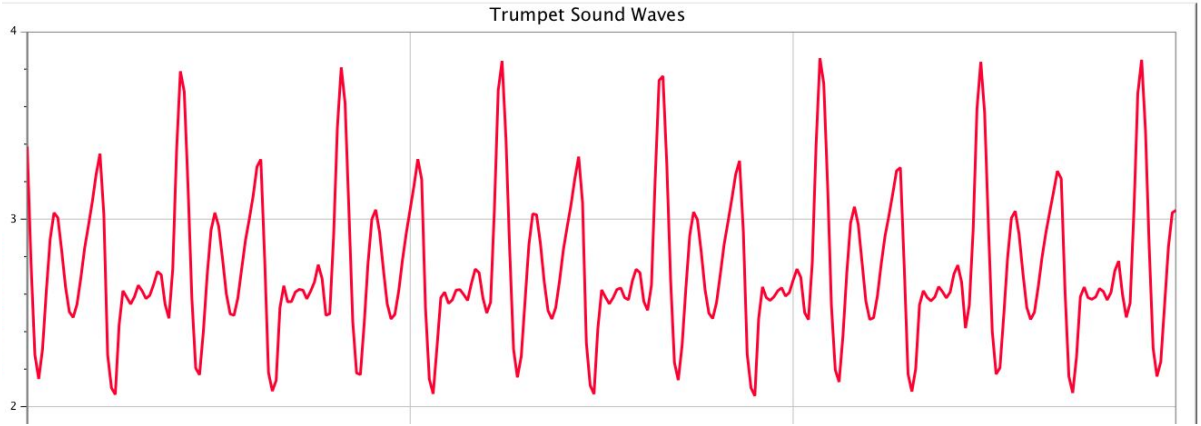
Audio Waves

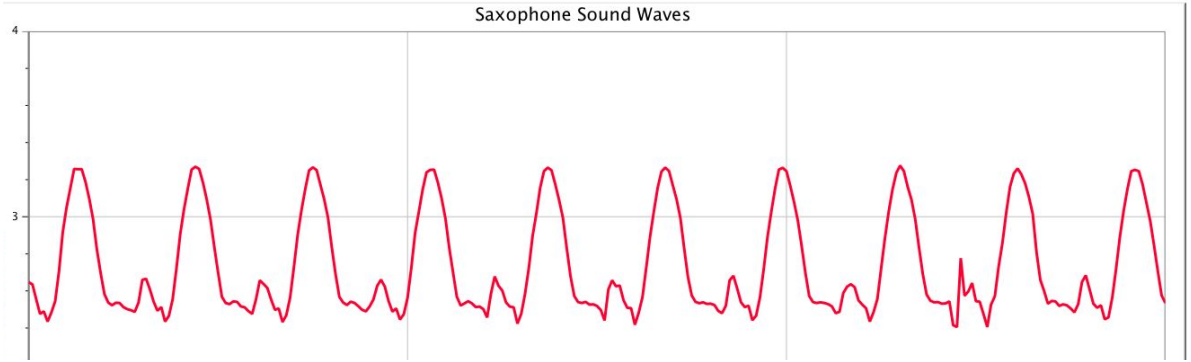






Audio Waves

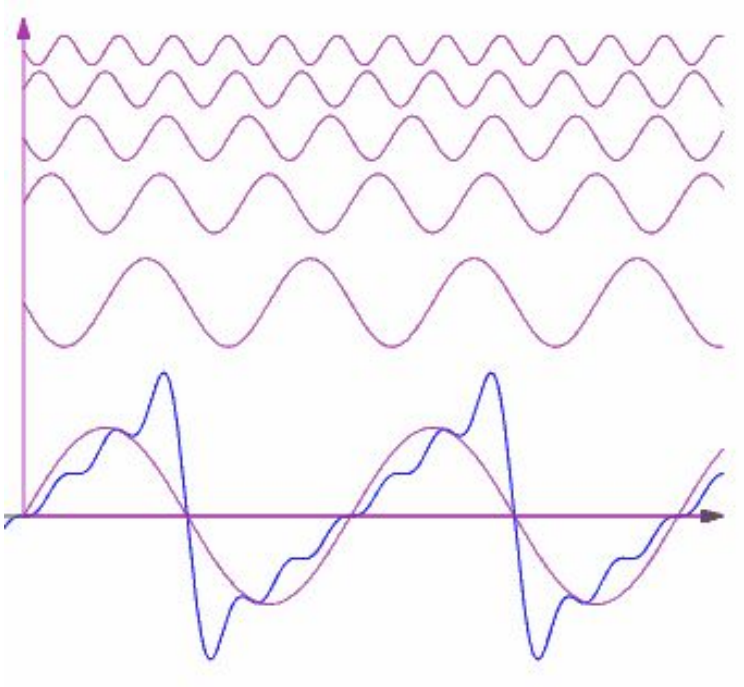


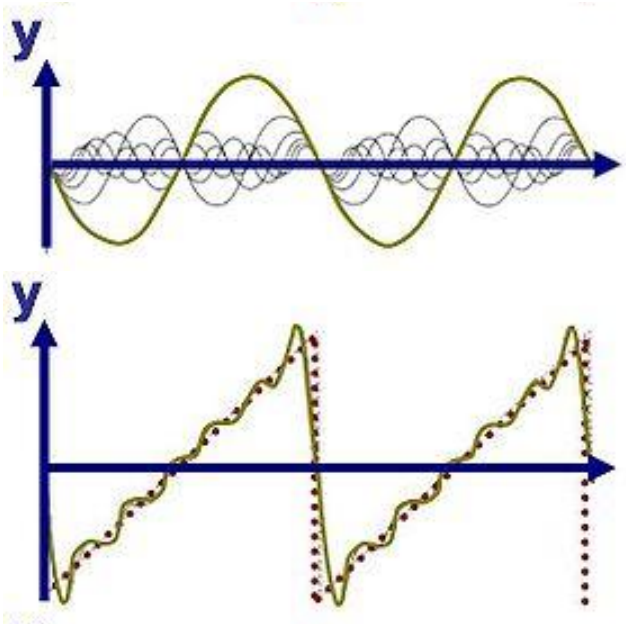


Audio Waves Processing

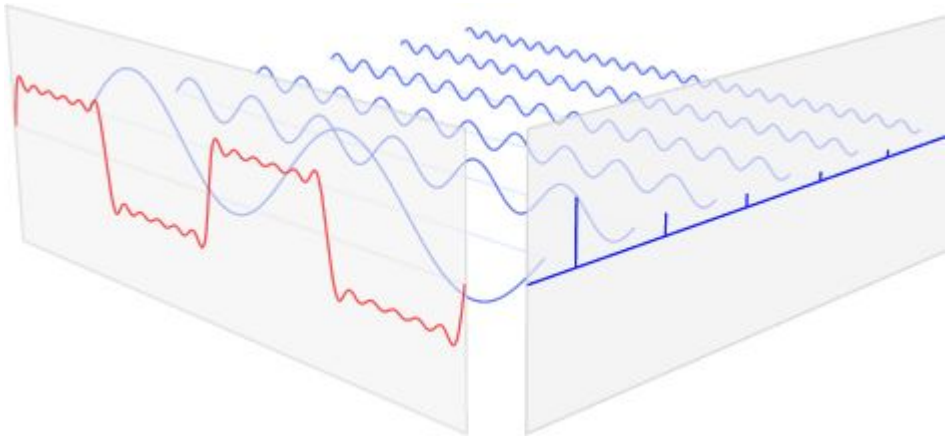
```
void setup() {
  size(700, 400);
  minim = new Minim(this);

  // set up an output channel
  out = minim.getLineOut(Minim.MONO, width);

  // init an oscillator and patch it to output
  wave0 = new Oscil(440, 0.3f, Waves.SINE);
  wave0.patch(out);

  wave1 = new Oscil(666, 0.3f, Waves.SINE);
  wave1.patch(out);

  wave2 = new Oscil(570, 0.3f, Waves.SINE);
  wave2.patch(out);
}
```

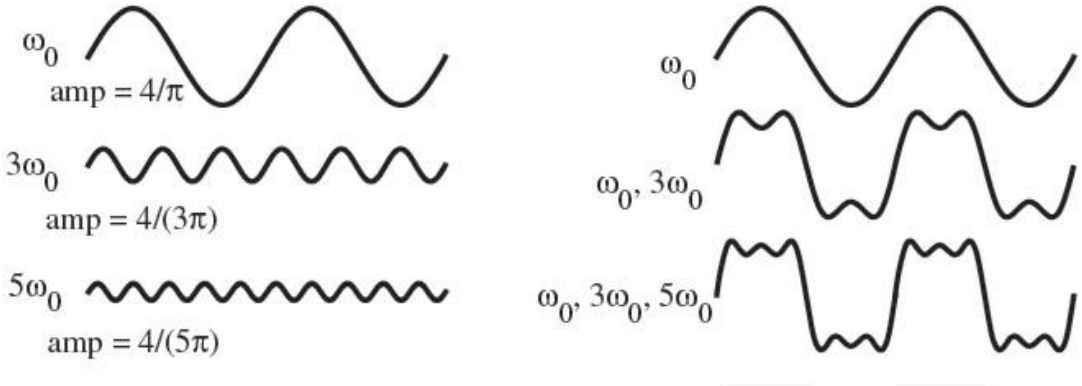
Fourier Theorem





Fourier Theorem

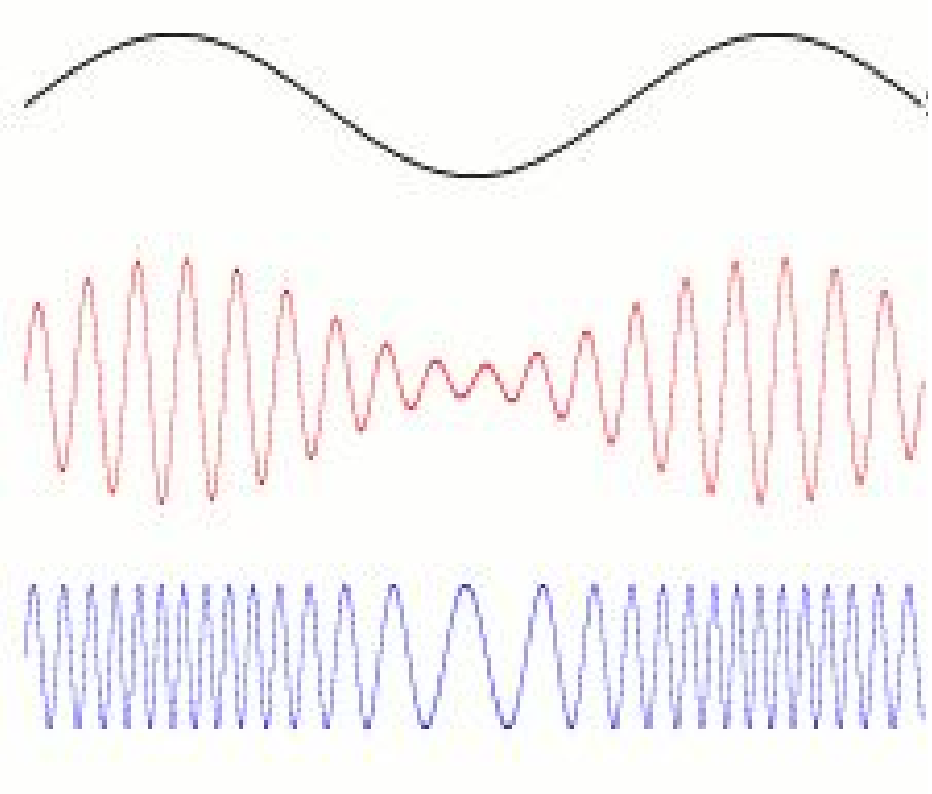




Minim Instruments

```
class ToneInstrument implements Instrument {
  Oscil sineOsc;

  ToneInstrument(float frequency, float amplitude) {
    sineOsc = new Oscil(frequency, amplitude, Waves.TRIANGLE);
  }

  void noteOn(float dur) {
    sineOsc.patch(out);
  }

  void noteOff() {
    sineOsc.unpatch(out);
  }
}
void keyPressed() {
  out.playNote(0, 0.2, new ToneInstrument(800, 0.8));
}
```

Shaping Waves: Low-Frequency Modulation

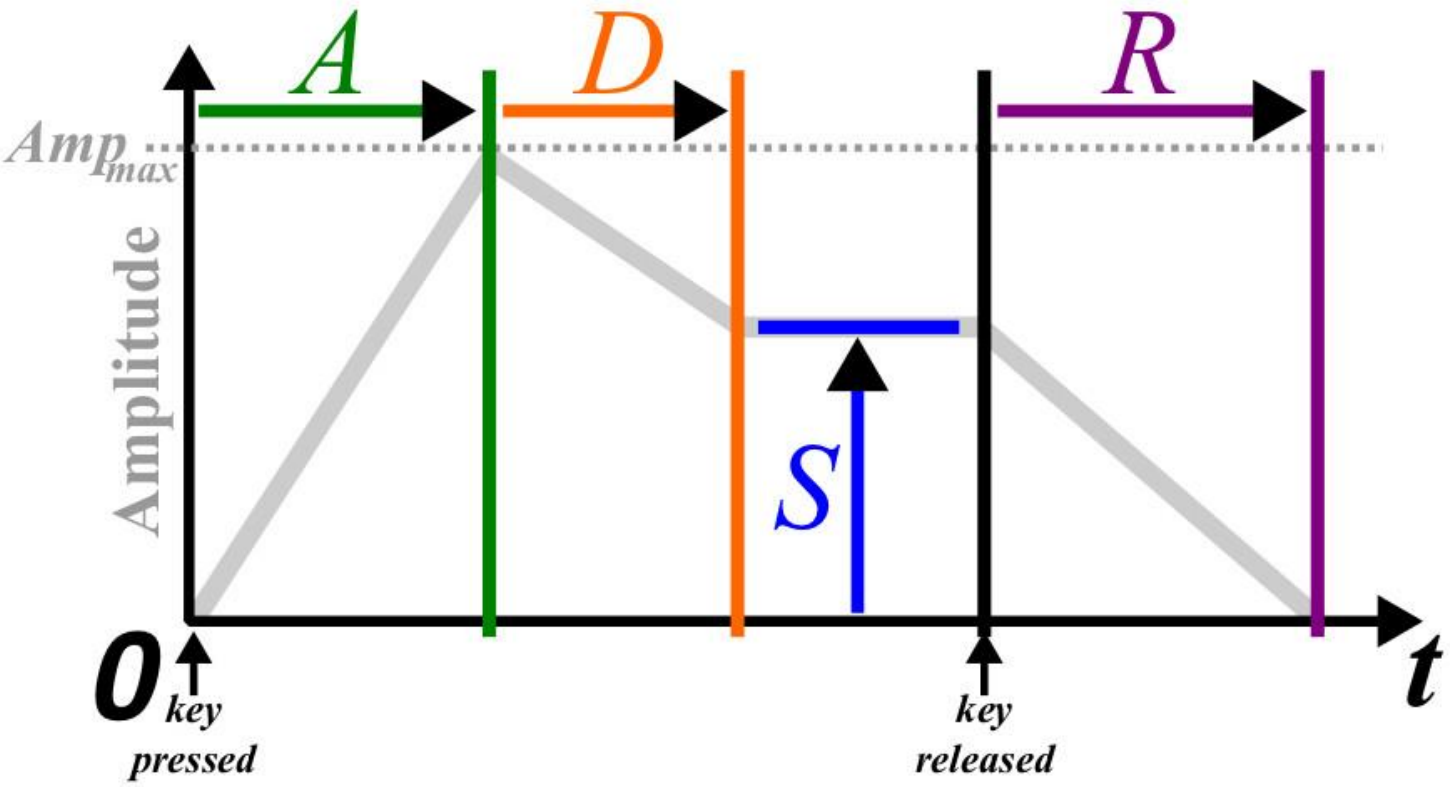


Original Signal

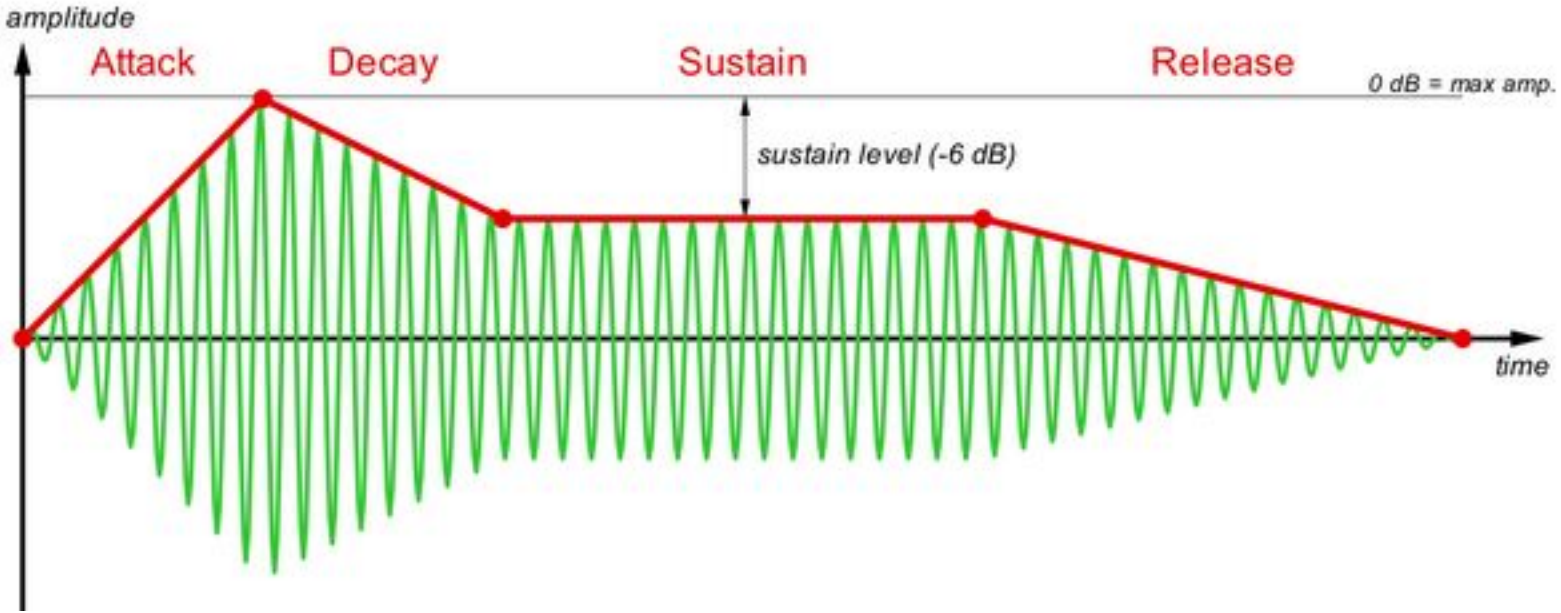
Amplitude Modulation

Frequency Modulation

Shaping Waves: Attack Decay Sustain Release



Shaping Waves: Attack Decay Sustain Release



Pixel

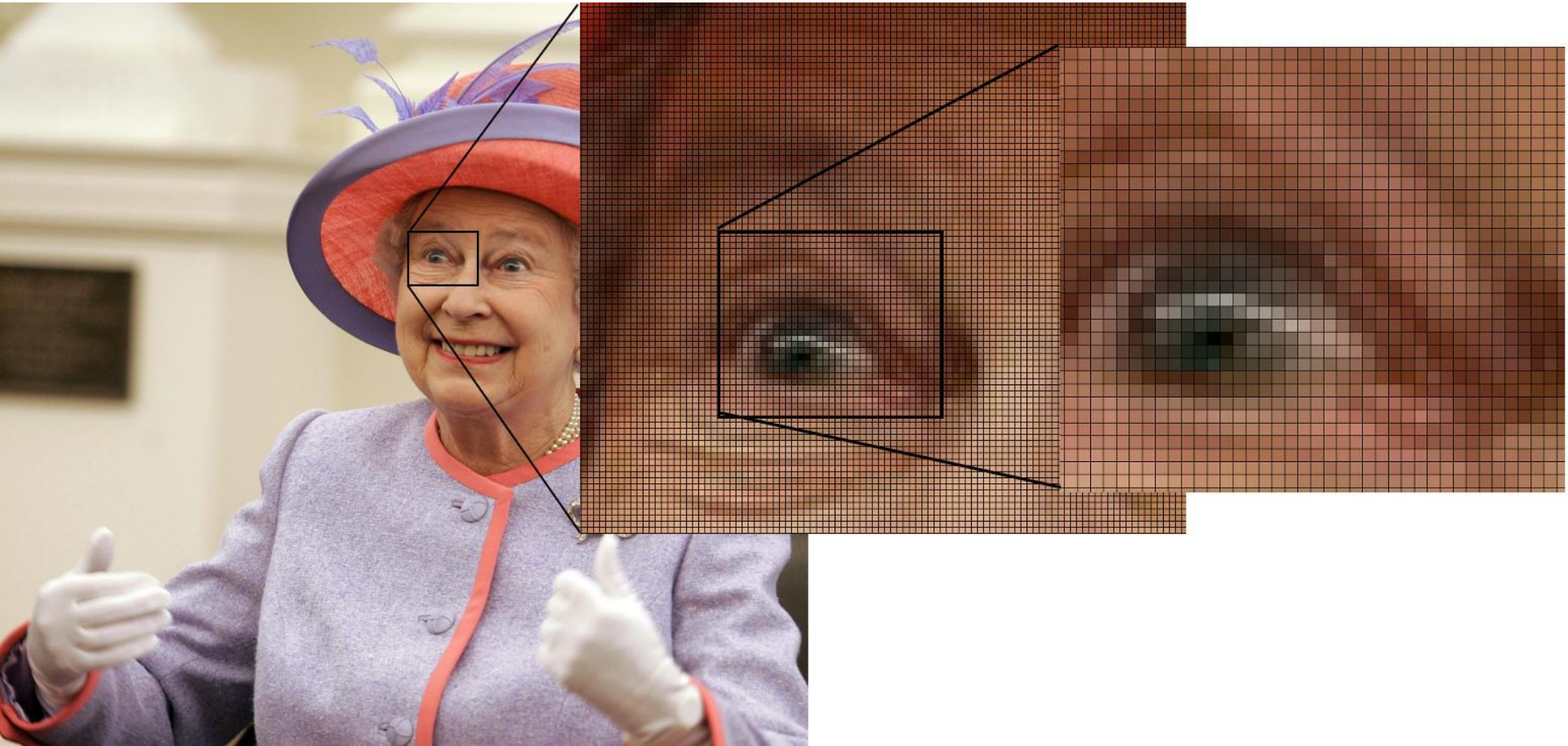


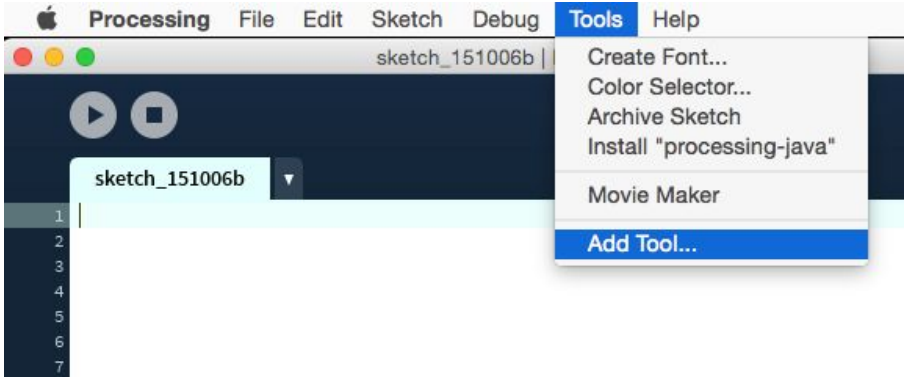
Image Processing ... in Processing

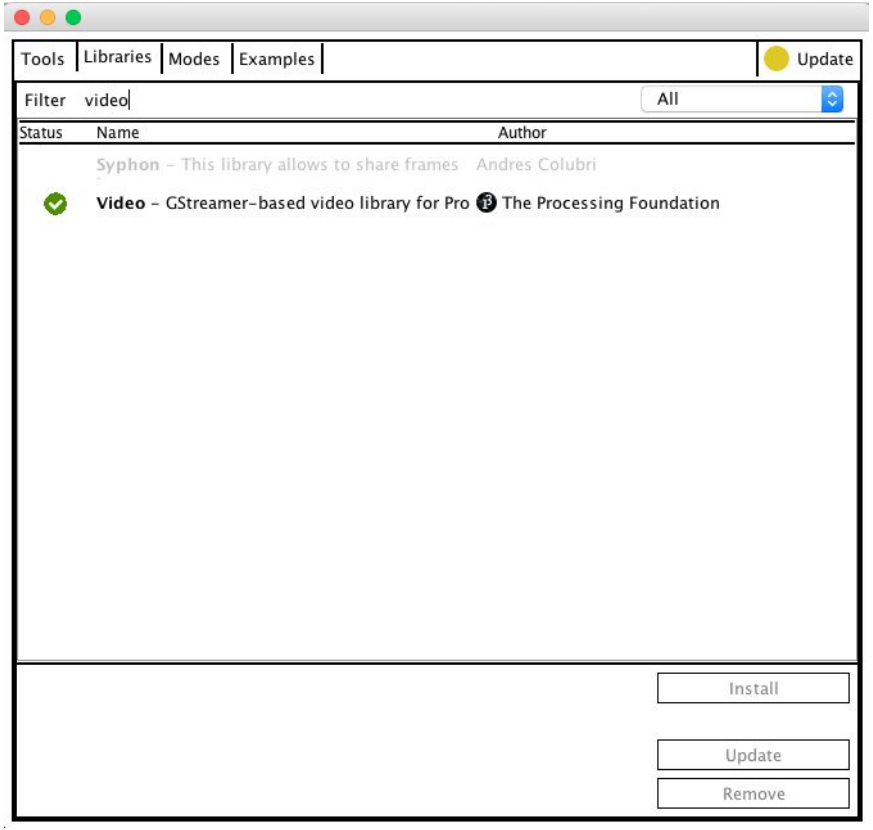
```
PImage myImage;

void setup() {
  size(720, 720);
  myImage = loadImage("queen.jpg");
}
void draw() {
  image(myImage, 0, 0);
}
```

Image Processing ... in Processing

```
for (int y=0; y<myImage.height; y+=1) {
  for (int x=0; x<myImage.width; x+=1) {
    // do something here with pixels
  }
}
```

Video Processing ... in Processing





Video Processing ... in Processing

```
import processing.video.*;

Capture myVideo;

void setup() {
  size(640, 480);
  myVideo = new Capture(this, 640, 480);
  myVideo.start();
}

void draw() {
  if (myVideo.available() == true) {
    myVideo.read();
  }
  image(myVideo, 0, 0);
}
```

Video Processing ... in Processing

```
myVideo.loadPixels();
for (int y=0; y<myVideo.height; y+=1) {
  for (int x=0; x<myVideo.width; x+=1) {
    // do something with pixels here
  }
}
```