Granular synthesizer

Made in the Web with the help of the Web Audio API

Panagiotis Katsos

January 26, 2022

Athens University Of Economics And Business
Multimedia technology

Table of contents

1. Introduction
2. Effects
3. Architecture and Development
4. Demo !!
5. Second thoughts and conclusion

    Resources

Introduction

What is it ?

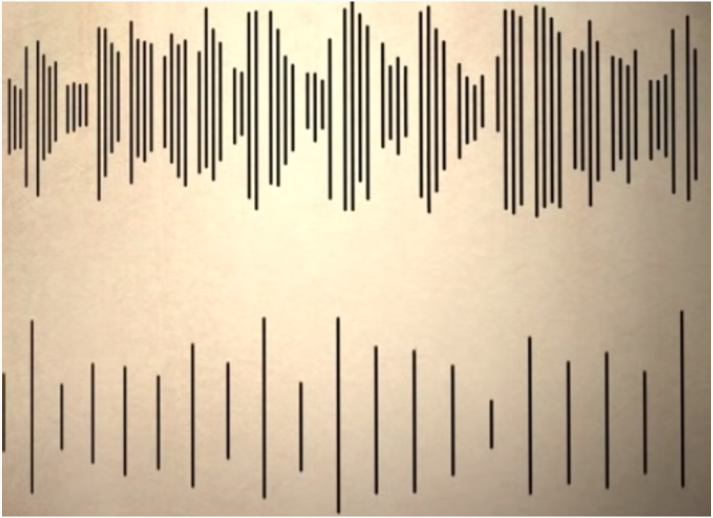
the Granular Synthesizer its an experimental way of making music.
it is mainly used in electronic music
it provides limitless possibilities
it is a way to push the limits of sound

What does it do ?

the **process** goes something like this.

- The user enters the audio of his choice.
- the audio is deconstructed into grains
- and reconstructed again in a random order

Decomposition



Recomposition

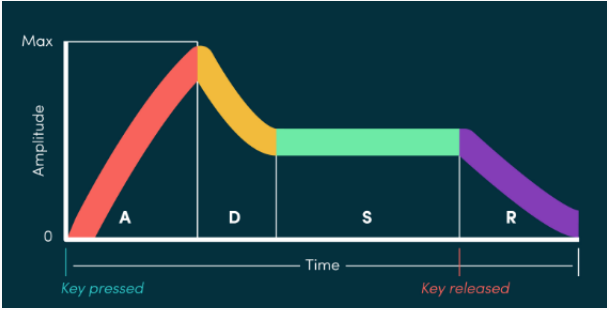
Effects

Why effects?

Effects are the most crucial part in the sound design process

- they shape the sound
- Unwanted clipping and annoying frequencies
- Achieving more coherent and concrete sound
- Adding creativity

ADSR envelope



Envelopes help us shape the time programming of a sound

- in my app it only makes sense to use attack and release
- attack : how much time does it take to reach the maximum volume
- release : how much time does it take to reach the 0 volume

Reverb & Delay

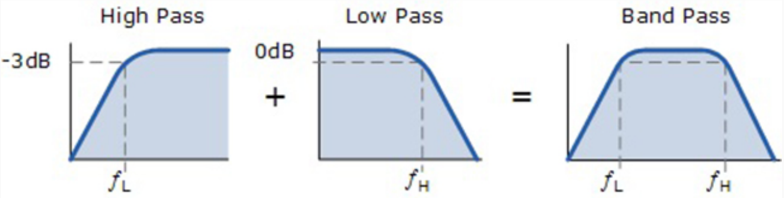
Reverb gives space to our sound

- we can specify a lot of parameters!
- Wet: how much percentage of the signal gets processed
- Decay: how much time the tail takes

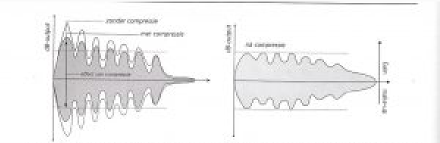
Delay repeats a part of the signal in specific time intervals

- similar to an echo
- ping pong delay
- Wet: again!
- DelayTime: duration of each delay
- feedback : how much of the signal is fed back

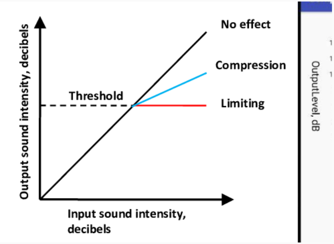
Filter



compressor



Figuur 6.11. Voorstelling van compressie op een audiosignaal. Links: het signaal vóór de bewerking; rechts: signaal ná de compressor-bewerking is gelijkmatiger en harder uit te sturen met de Gain make-up structuur



the **compressor** squashes the audio signal and gets rid of volume peaks.

- knee : percentage of compression
- db : in which volume the effect is being applyed

Phaser

the **phaser** is an experimental effect

- creates noches typicaly on the lowest of frequencies
- modulates them with an LFO
- phaserFrequency the frequency the noches are centered around
- octaves the range of the effect
- GIF

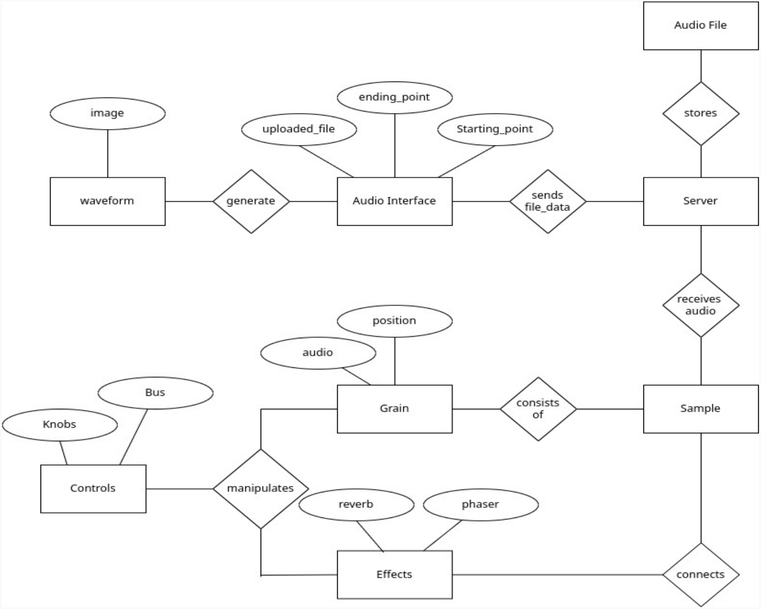
Architecture and Development

General

The **architecture** is mainly client server.

- the frontend has been build with React
- the audio processing has been done with tonejs
- the server side has been developed with nodejs and express

Overview


Audio File
stores
image
ending_point
uploaded_file
Starting_point
waveform
generate
Audio Interface
sends file_data
Server
receives audio
position
audio
Bus
Knobs
Grain
consists of
Sample
Controls
manipulates
reverb
phaser
Effects
connects

Demo !!

Second thoughts and conclusion

Problems, problems, problems

I found this project to be more demanding and I encountered a lot of problemsjk

- nodejs does not support the web audio api
- very little amount of examples of similar projects
- tonejs has very limiting documentation
- React is awesome but it has its limitations

Lessons

this was by far the **longest** and **hardest** project I made so far, so I learned a lot.

- learned how to structure a big project from the start
- keeping my code clean
- learning frameworks better
- learned how to be more resourceful

Links

- UI : React
- API : nodejs
- audio processing : tonejs
- waveform creation : waveform-generator-web
- drag and drop list: React DnD
- animation : React spring
- mp3 cropping : mp3 cutter
- knobs: jQuery knob
- bootstrap : react bootstrap
- LaTeX Beamer Theme: METROPOLIS