

Fragment

The Collaborative Spectral Synthesizer

by Julien Verneuil - contact@fsynth.com

This is the documentation of the Fragment synthesizer.



(<https://www.fsynth.com>)

Table of Contents

Contents

1	Introduction	1
1.1	History	2
1.2	Capabilities	2
1.3	System requirements	3
1.3.1	Browser	3
1.3.2	CPU	3
1.3.3	GPU	3
2	Concepts	4
2.1	Additive synthesis	4
2.2	The graphical score	4
3	Sessions	5

1 Introduction

The **Fragment synthesizer** (<https://www.fsynth.com>) (also called fsynth) is a collaborative web-based musical instrument which allow direct manipulation of the sound spectrum by the use of on-the-fly GPU (Graphics Processing Unit) programming.

Fragment is stereophonic, polyphonic, multitimbral and support live coding of audio and visuals at the same time.

Fragment is first and foremost a powerfull additive synthesizer which let you have complete control over the sound spectrum in real-time with direct visual feedback.

Many videos of Fragment live sessions were recorded and are available on YouTube as a playlist (https://www.youtube.com/playlist?list=PLYhyS20KJmge_PEImydWZN1KbvCzkjgeI).

1.1 History

In 2009, i discovered the Virtual ANS (<http://www.warmplace.ru/soft/ans/>) synthesizer by Alexander Zolotov (<http://www.warmplace.ru>), a software emulation of the Russian photoelectronic synthesizer ANS (http://en.wikipedia.org/wiki/ANS_synthesizer) which was created by the Russian engineer Evgeny Murzin (http://en.wikipedia.org/wiki/Evgeny_Murzin) from 1938 to 1958.

I was blown away by the remarkable possibilities offered by the Virtual ANS which let you draw the sound spectrum over time, i then discovered programs such as MetaSynth or HighC (<https://highc.org>), this is how i started to experiment with the "drawn sound" method.

Fragment started in 2015, when i was making the first prototype of a web-based ANS-like synthesizer software which is still a work in progress, a prototype of Fragment was made in a single week, the prototype was quite mature but lacked in features, i considered Fragment as a pretty but failed experiment at the time.

In the summer 2016 while i was releasing the prototype source code, i played with it again and after tweaking the GLSL code, i was able to make some nice sounds and envision what would be possible with this technology, so i started to think about that prototype again and come up with many ideas that would make Fragment more functional and easier to use, the real work begun.

After many hours spent crafting the software, the first version of Fragment was released in January 2017, it was quite limited at that time, Fragment has now improved alot.

1.2 Capabilities

- Powerful additive synthesizer
 - Powered by WebAudio oscillators
 - Powered by FAS, an independent program
 - Powered by a wavetable
- Stereophonic
- Monaural
- Polyphonic
 - Automatically detected from the GPU capabilities
 - 16 notes minimum
 - 704 notes with a GeForce GTX 970 GPU
- Multitimbral
- Spectral recording with mixing and export as PNG
- Adjustable audio output channel per slices
 - Multiple audio channels are only supported with FAS
- Shader inputs, webcam, textures, audio files
- MIDI Enabled
 - Only with WebMIDI API enabled browsers (Google Chrome)
 - Hot plugging of MIDI devices is supported
- Collaborative app.
 - MIDI and shader inputs are not synchronized between users
- Feedback support
- Live coding/JIT compilation of shader code

- Global and per sessions settings automatic saving/loading
- No authentications (sessions based)
- Per-sessions discussion system

1.3 System requirements

Fragment is a special kind of additive synthesizer which require a moderate deal of processing power in order to work properly, a medium-end GPU and CPU should work fine.

1.3.1 Browser

Fragment require a browser with full support for ECMAScript 5, CSS3, WebAudio and WebGL.

Well-tested and compatible browsers include Firefox 51.x and Chrome 55.x but Fragment may work with previous version of those browsers as well.

It is known to work on recent version of Safari and Opera as well.

Chrome or Chromium browser is recommended.

Fragment support the WebGL 2 API which improve performances and enable some advanced features in the fragment shader due to GLSL 3.0 usage.

1.3.2 CPU

Fragment may be quite hungry in term of computing resources, a dual core with high clock speed is recommended, a CPU with more cores can be useful if you just want to use the browser for audio output.

Several methods are provided to synthesize sounds, each with performances pros and cons:

- **FAS (recommended):**
 - Very fast
 - Multiple output support with soundcard choice
 - Many settings
 - Can run on a dedicated computer such as a Raspberry PI
 - Dedicated program which receive Fragment data over the network
 - Enhanced noise modeling by adding band-limited noise
- WebAudio oscillators:
 - Fastest under Chrome
 - May not work with Firefox
 - Require a fast CPU
 - Integrated global "release" envelope
- Wavetable (not recommended):
 - Most compatible browser method
 - Will produce crackles

1.3.3 GPU

Fragment was developed and tested with a NVIDIA GeForce GTX 970 GPU, a powerful GPU may be required if:

- you want higher score resolution
- you want to do visuals alongside audio

- you are doing complex things/use many inputs in your shader
- you want greater polyphonic/harmonics/partial capabilities

2 Concepts

Fragment is an additive synthesizer, it let you have full control over the timbral qualities of your sounds by the mean of GPU programming.

Unlike other additive synthesizer software, there is no need for knobs, sliders or any other controllers to sculpt your sounds, all of that is done by generating visuals which will determine your sounds harmonic content and dynamic characteristics.

There is many features in Fragment which allow any creative minds to produce complex sounds in different ways and even with external tools, i suggest all the readers to look at great softwares like the IanniX sequencer (<https://www.iannix.org>) and to use it with Fragment using the MIDI capabilities.

While Fragment may seem overwhelming at first, the only thing that is required to produce sounds with it is to know how to generate the visuals.

The goal of this section is to clarify the inner working of Fragment.

2.1 Additive synthesis

Fragment is first and foremost a powerful additive synthesizer which make an extended use of additive synthesis.

Additive synthesis is a sound synthesis technique that creates timbre by adding sine waves together.

Adding sine waves produce a timbre, the timbre quality is mainly defined by its harmonic content and the dynamic characteristics of the harmonic content.

The concept of additive synthesis is centuries old, it has first been used by pipe organs.

Fragment can theoretically produce any timbres with precise accuracy.

The only limit to the amount of sine waves that can be added by Fragment is the limit of the available processing power.

For example, on a Raspberry PI 3 (1.2GHz 64-bit quad-core ARMv8 CPU) 700 oscillators can be played simultaneously using two cores, matching the capability of the ANS synthesizer.

Actually, there is a also a hard limit which is that the frequency capture of slices cannot go beyond the refresh rate of your monitor (and this also depend on the browser implementation), this could however be bypassed easily by switching to a different monitor.

Fragment can also do other types of synthesis like subtractive synthesis.

2.2 The graphical score

Fragment graphical score represent the sound spectrum which is generated by the GPU from a fragment program.

The fragment program (also known as a fragment shader) is executed by the GPU and compute the color and other attributes of each "fragment" - a technical term which usually mean a single pixel.

The graphical score represent a kind of sonic canvas where the X axis represent time and the Y axis represent frequencies, you "paint" the graphical score by writing a fragment program which

will be executed for each pixels of the graphical score.

What you hear in Fragment is determined by the position of "slices" which are added on the graphical score, slices are vertical bits of the graphical score which are merged together and produce an audible result.

The content of slices is captured at the rate of display refresh rate which conventionally should be 60fps most of the time.

The frequency mapping of the graphical score is fixed by a logarithmic formula, although the formula cannot be changed right now, some parameters are available in the settings dialog to fine tune the frequency map.

The frequency mapping is defined by the formula:

$$f(y) = a * (2^{\frac{y}{n/o}})$$

Where:

- **a** is the starting frequency
- **y** the vertical position
- **n** the number of oscillators (which is the height of the canvas)
- **o** the octave count

3 Sessions

Fragment sessions are isolated spaces which are open to all peoples who has access to the session name, they can be joined by going to the Fragment homepage (<https://www.fsynth.com>) or by typing the session name directly into the address bar as demonstrated below.

You can join any sessions directly from your browser address bar by replacing "yoursessionname" for the URL shown below by the session name you want to join or create, session names are limited to a maximum of 100 characters and cannot have the "[Error: Lexical error in '/home/julien/Projets/fs/documentation/documentation at 180:249[11514]': \$" at line 179]