THREE AUDIO VISUALIZER

NGO LAP NGUYEN

Overview

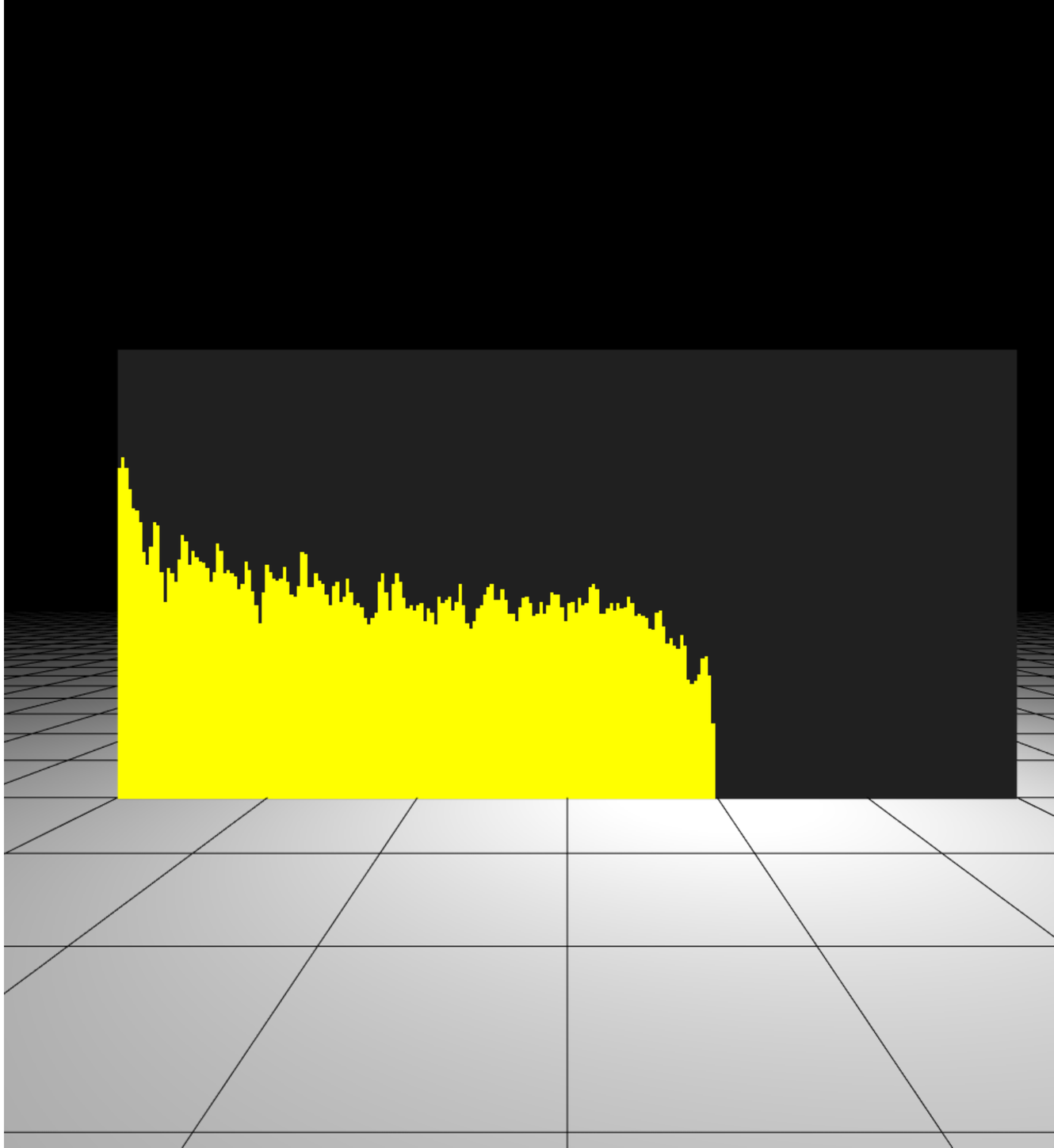
Heads-up

This is not a *guide*, but a *sum-up* of what I've learned.

About the app

Basic Audio Visualizer built with

- ThreeJs
- WebAudio API
- React (react-three-fiber)



Content

1. **Basic ThreeJS concepts**
    a. Introduction & Sample projects
    b. Basic constitution of a ThreeJS scene

2. **Basic WebAudio API concepts used in the project**
    a. Fast Fourier Transform
    b. AnalyserNode

3. **General Flow – Process & Display the Audio**

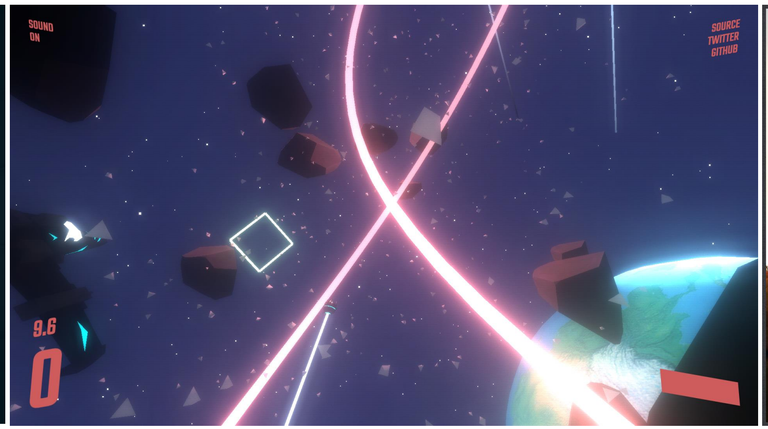
Introduction & Sample Projects

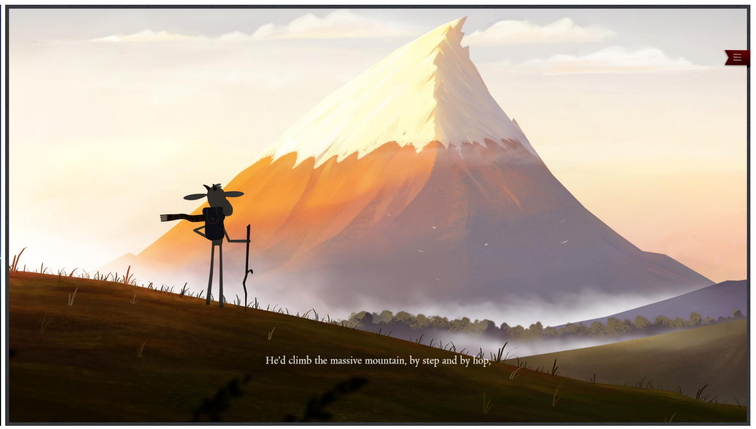
A cross-browser Javascript library and API used to create & display 3D graphics in a web browser.

A wrapper of WebGL, make it easier to use.

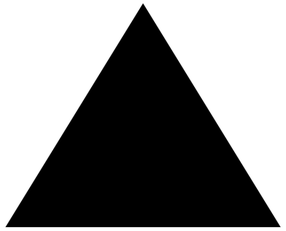
Author: Ricardo Cabello (mr.doob)

Initial Release: April 24th, 2010



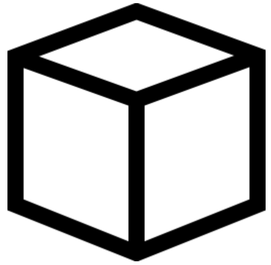


Basic constitution of a scene



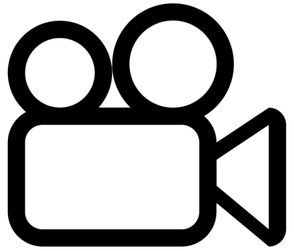
Vertex & Segment

The base building components in ThreeJS



Mesh

An object created by combining **Geometry** and **Material**



Camera

Just... Camera.

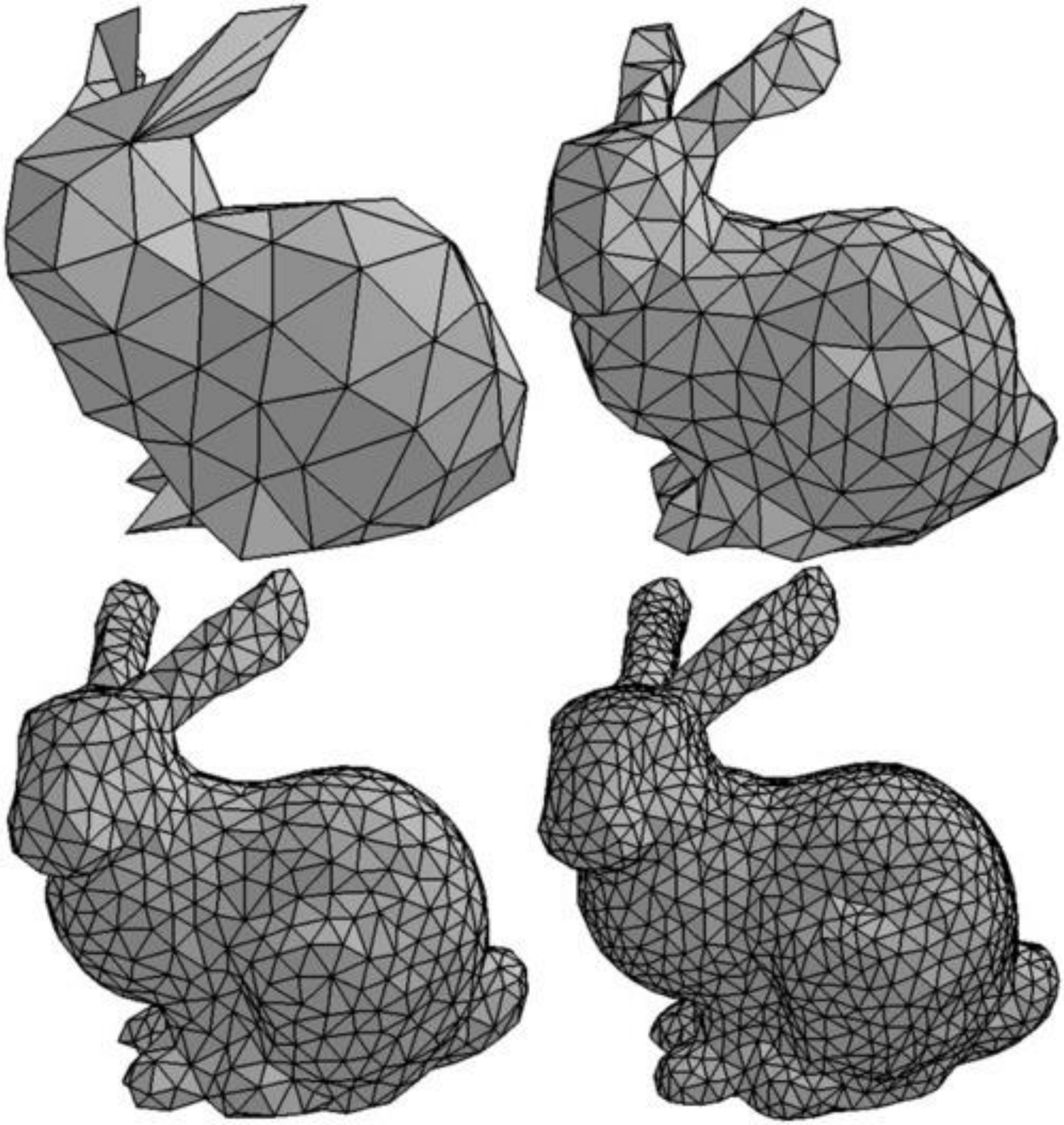
Light

Sources of light in the scene.

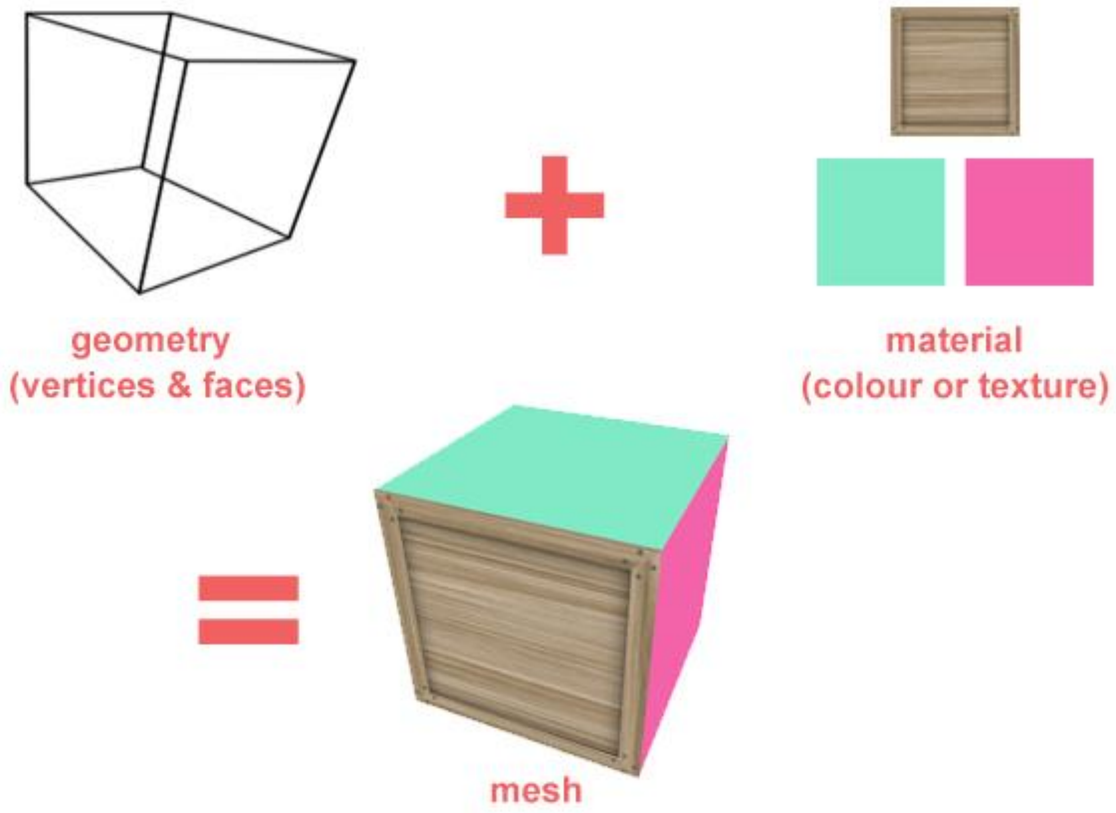
Vertex & Segment

Segment is the base building component in ThreeJS, created with **3 vertices** (basically it’s a triangle).

More vertices, more segments -> more details.



Mesh



An object created by combining geometry and material:

- **Geometry:** A collection of vertices and faces
- **Material:** The material you want the object to be made in

E.g: For a wooden box:

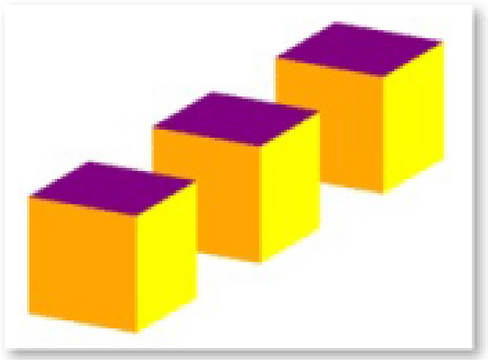
- Geometry: **Box**
- Material: **Wood**

Camera

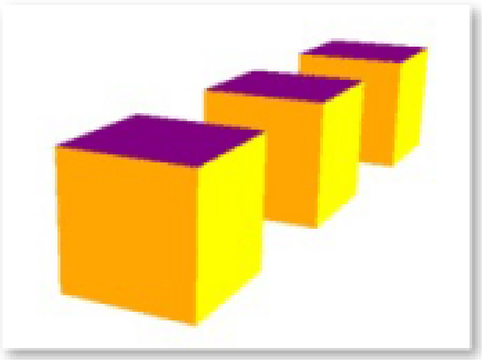
2 main type of cameras: **Perspective** and **Orthographic**

- Perspective camera is how we see the real world with **depth.** We can judge the distance.
- Orthographic camera remove that perspective (think of a 2D, side-scrolling game such as Mario).

Orthographic Projection



Perspective Projection

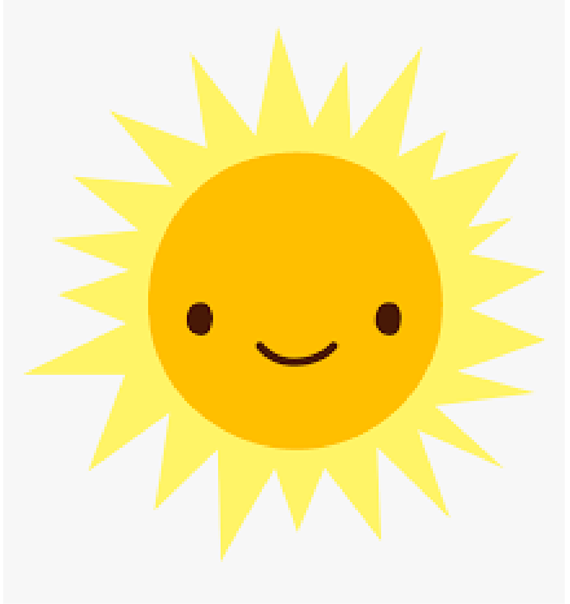


Light

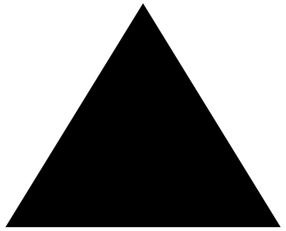
ThreeJS provides objects acting as different types of light source:

- **SpotLight**
- **PointLight**
- **AmbientLight**
- ...

Please refer to the examples for better illustrations.

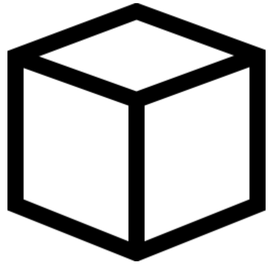


Basic constitution of a scene



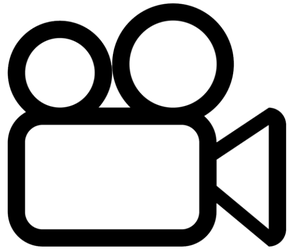
Vertex & Segment

The base building components in ThreeJS



Mesh

An object created by combining **Geometry** and **Material**



Camera

Just... Camera.

Light

Sources of light in the scene.

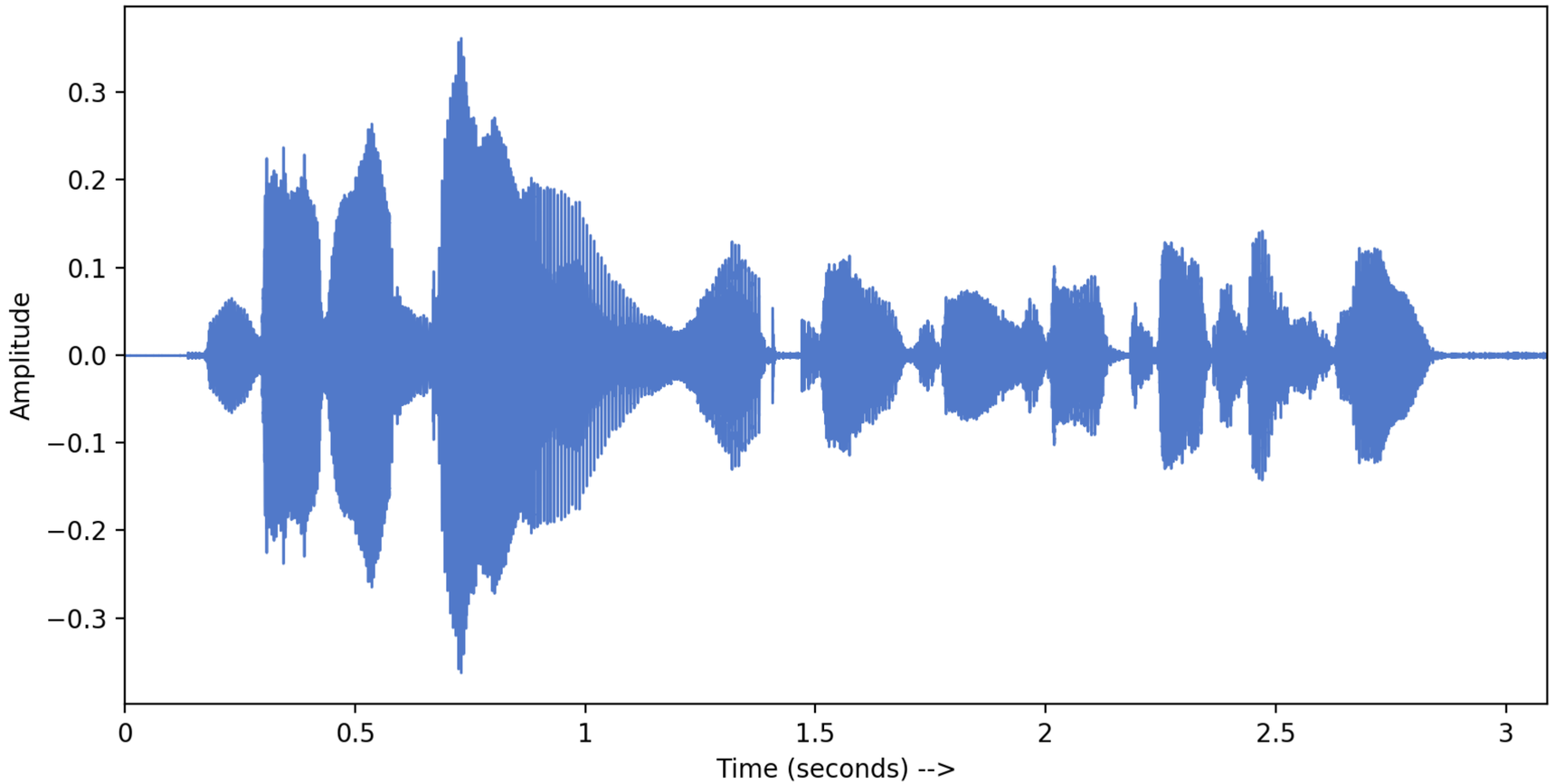
WebAudio API

"The WebAudio PI provides a powerful and versatile system for controlling audio on the Web, allowing developers to choose audio sources, add effects to audio, create audio visualizations, apply spatial effects (such as panning) and much more.“

Things I’ll go into:

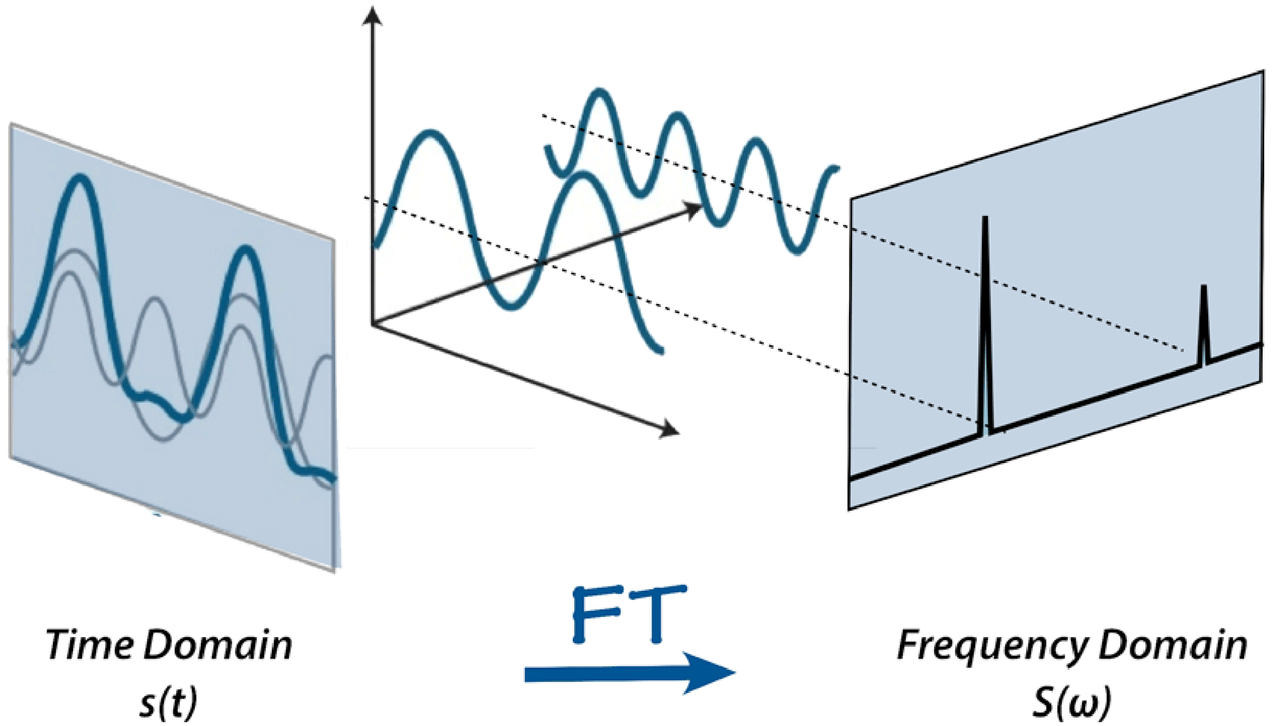
- Fast Fourier Transform (FFT)
- **AnalyserNode** from the WebAudio API

Fast Fourier Transform



Fast Fourier Transform

ELI5: Fast Fourier Transform is like a recipe finder.



AnalyserNode

We'll use **getFloatFrequencyData()** method, which uses FFT under the hood to determine the frequency domain. The returned value is a normalized array of values between 0 and 255, representing the power value (amplitude) of the sound at each calculated frequency.

For example:

- 1000Hz: 255 (loud),
- 2000Hz: 196 (less loud),
- 3000Hz: 40 (less less loud),
- 4000Hz: 0 (silent at this frequency),
- ...

GENERAL FLOW

Processing & Display

1. Input a sample audio file
2. Use the WebAudio API & Fast Fourier Transform (FFT) to convert the audio soundwave into frequency-domain data
3. Base on the data, render the shapes to visualize it with three.js

Q & A