

building mayk.it

from AudioKit to JUCE

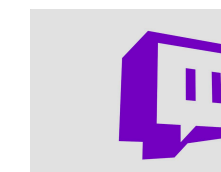
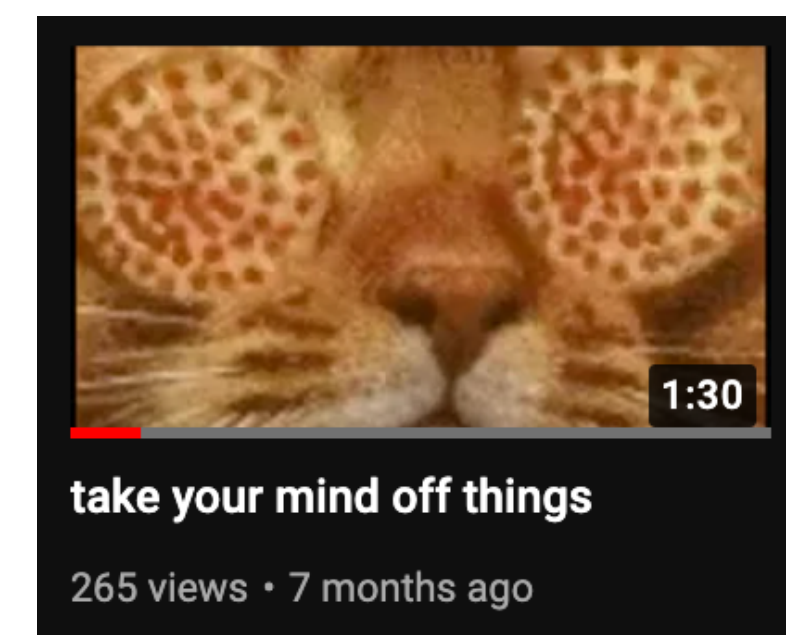
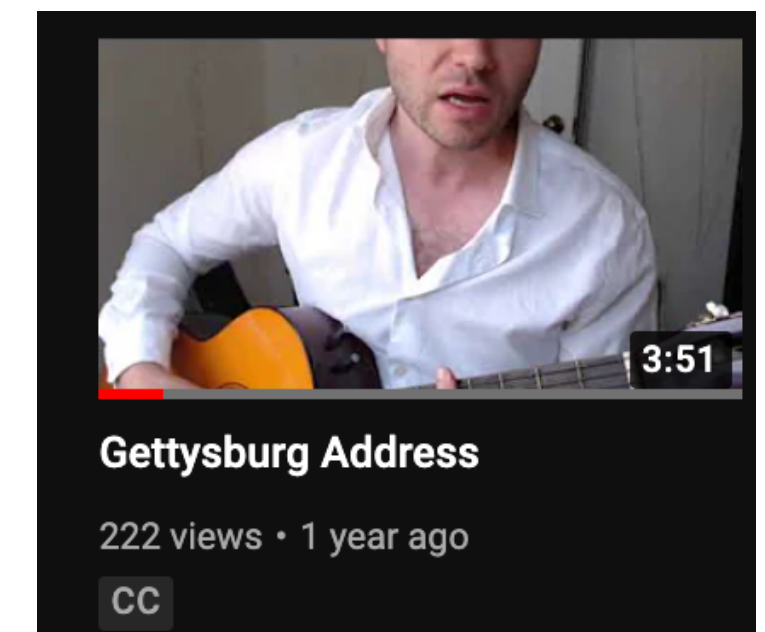
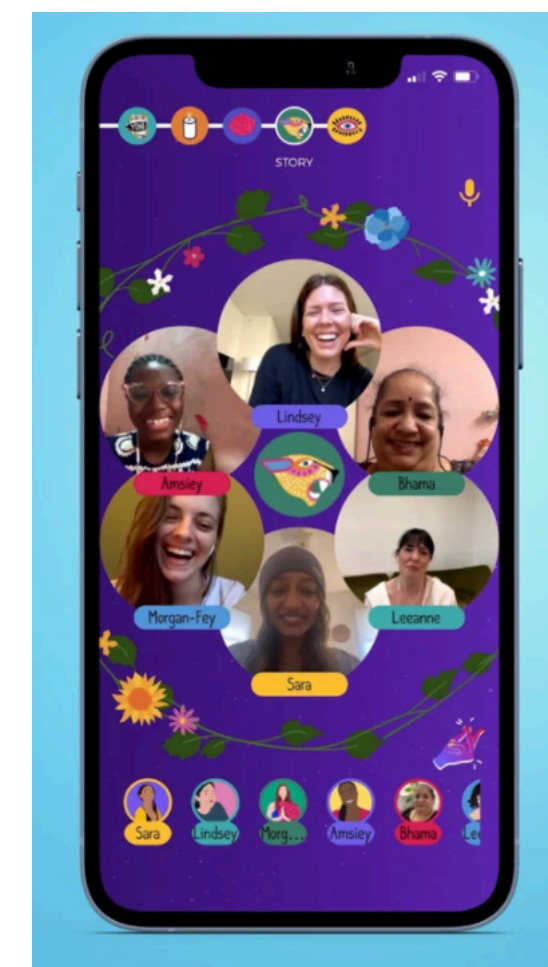
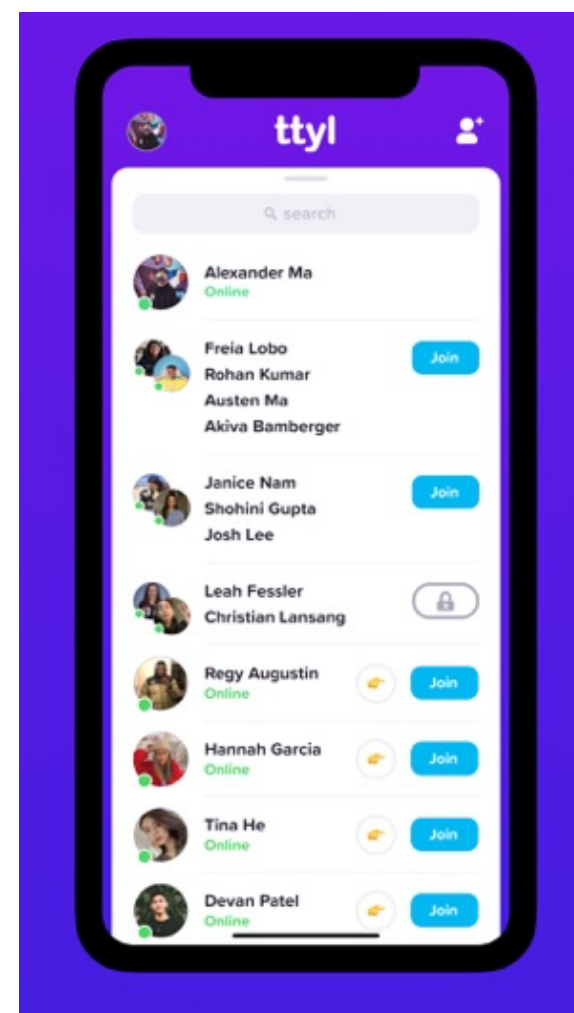


Akiva Bamberger, mayk.it ADC 2021

akiva@mayk.it

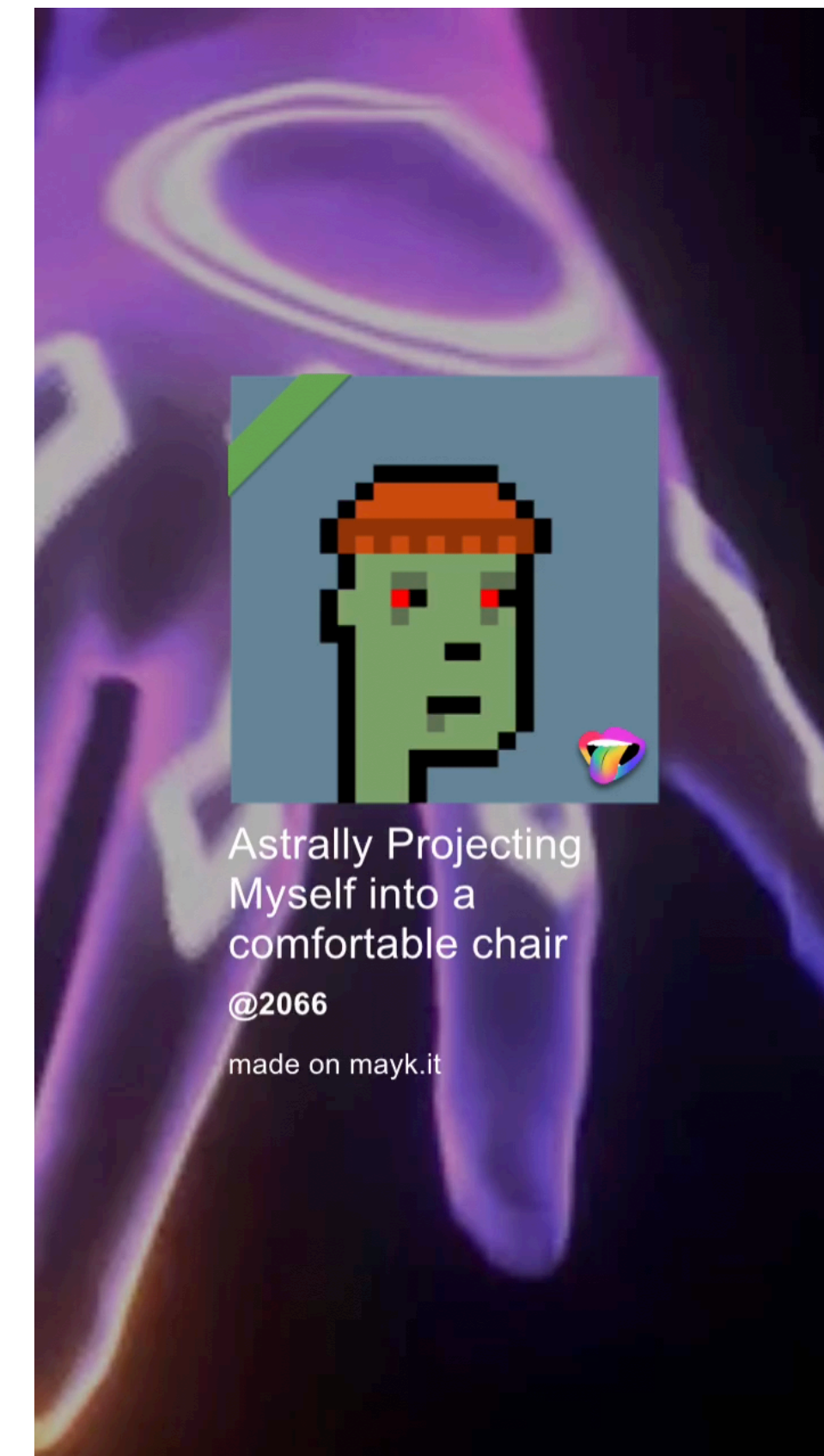
who am I?

- firmware engineer
- iOS engineer
- Folk singer, content creator (Youtube / Twitch)



mayk.it

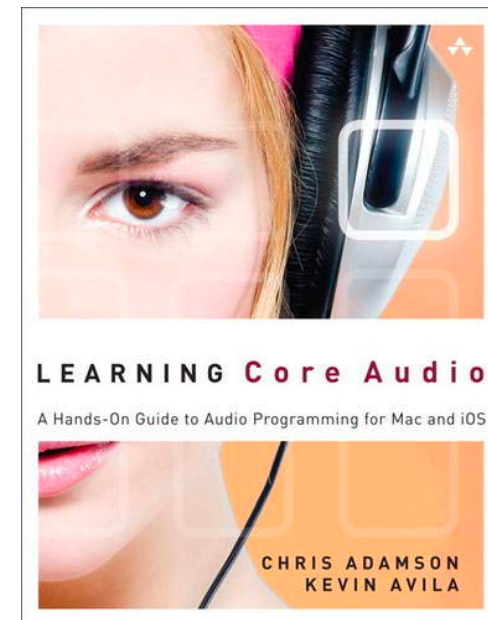
- “music app for non-musicians”
- Focus on beginner tools for mobile music making
 - Pitch correction
 - Produced beats
 - Community
 - Prompts / creative tools



audio for iOS



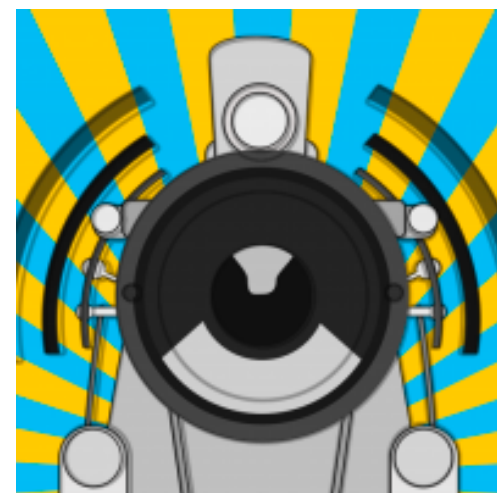
Core Audio, 2003



Learning Core Audio, 2012



iPhone, 2007



The Amazing Audio Engine, 2012 - 2016



Superpowered, 2013 (acq 2020)



AVAudioEngine, Sept 2014

AUDIOKIT

AudioKit v4, Sept 2017

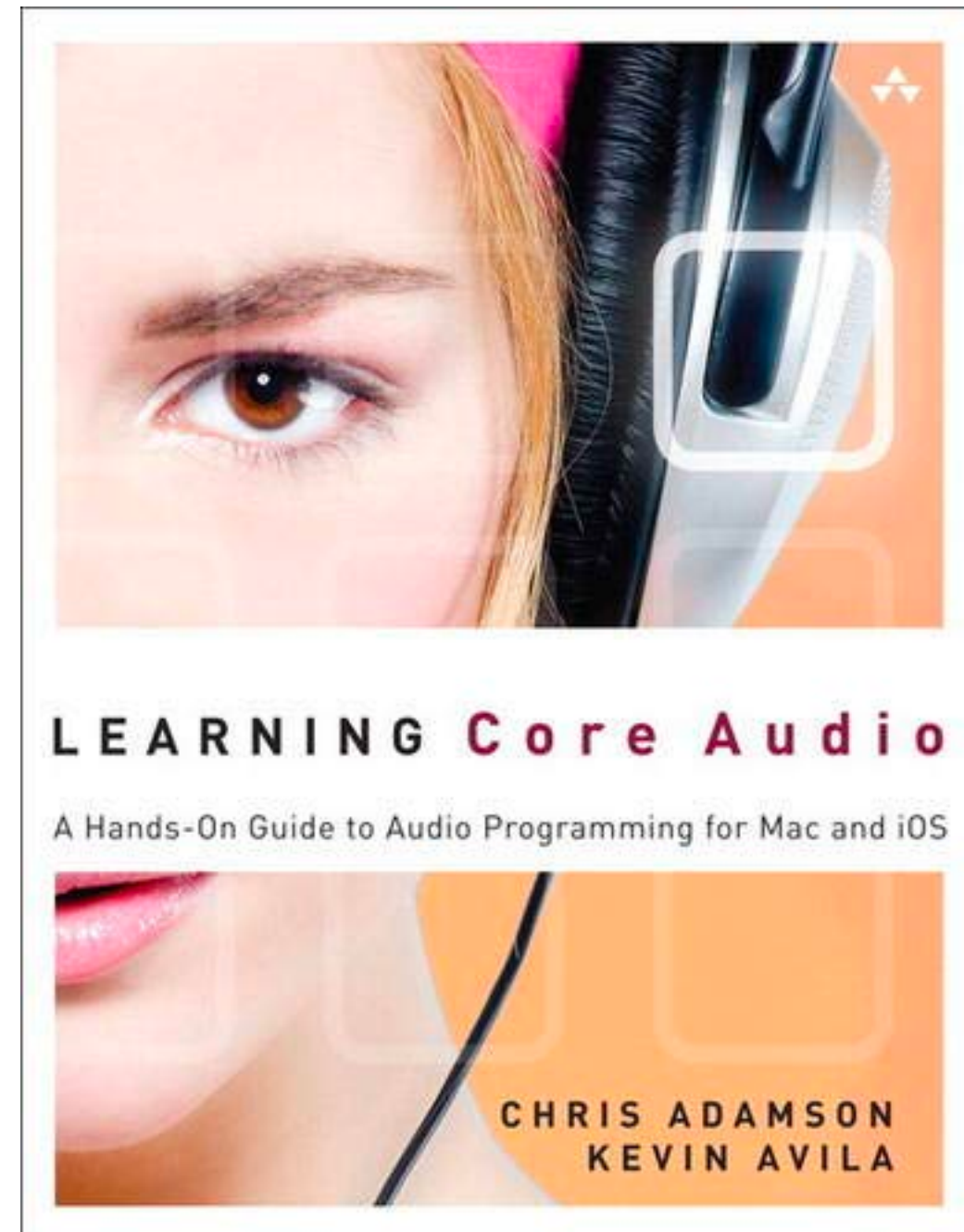


iPhone 6, 2014



AudioKit v5, 2020

Learning Core Audio, 2012



Learning Core Audio, 2012

“Problem with updating the book is that to address the relationship with AV Foundation and the new v3 Audio Units, we'd basically have to start over and write a whole new book -- **the organization of the old one doesn't make sense for the current media frameworks.** And given that I made about \$4/hour writing the last one once I divided my royalties by the time I put into it, I'm sorry to say a new mac/iOS audio book not at the top of my priority list at the moment.”

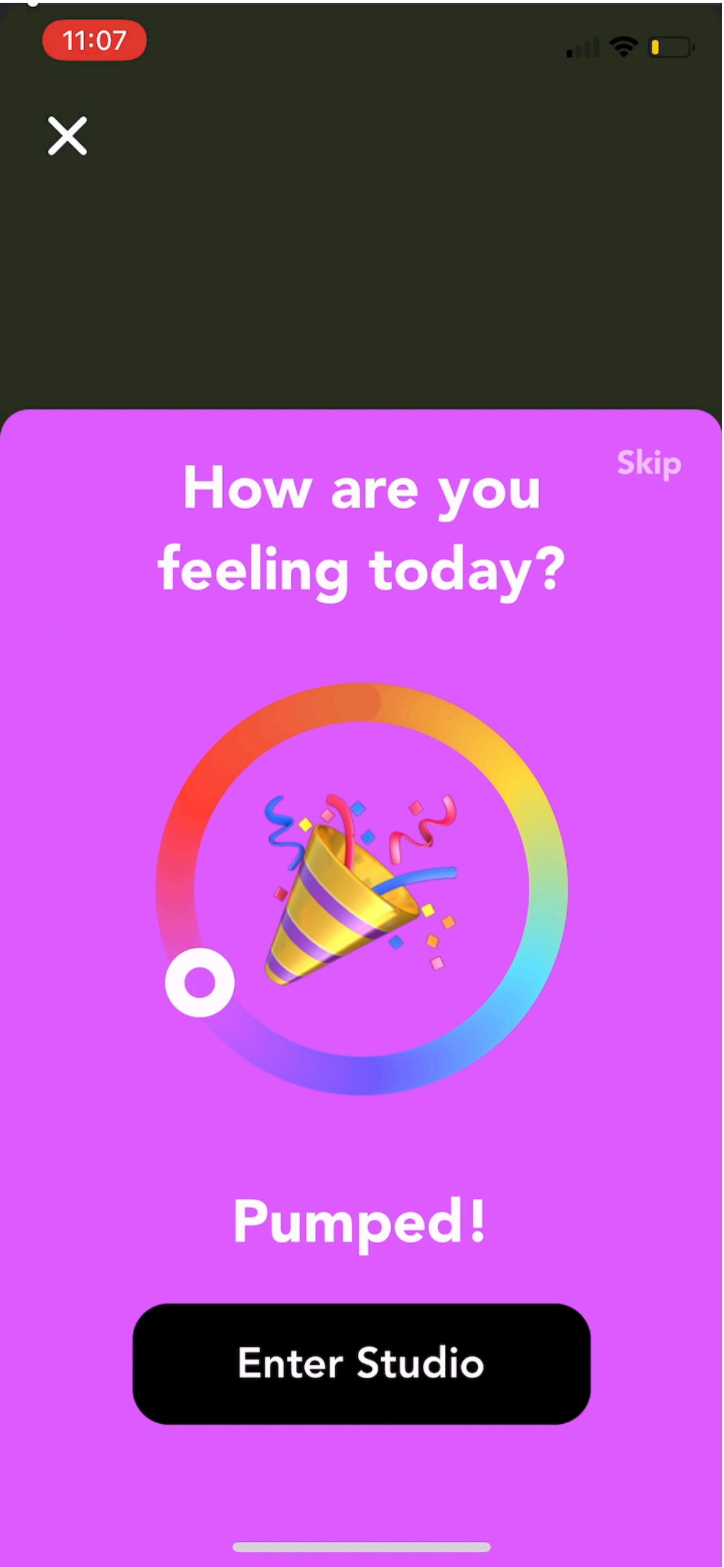
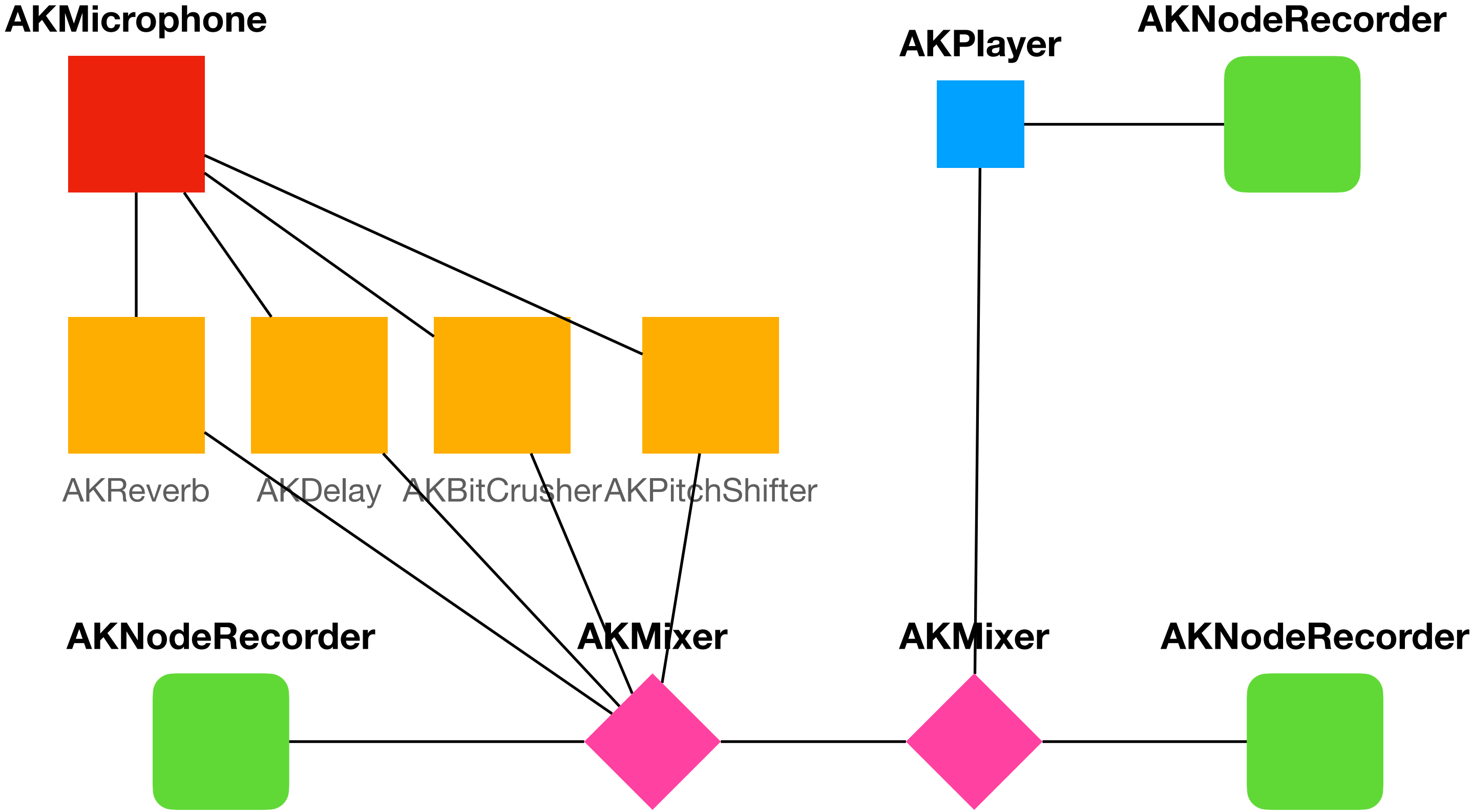
(From https://www.reddit.com/r/iOSProgramming/comments/4ttjln/learning_core_audio_2012_still_relevant/)

The Amazing Audio Engine, 2016

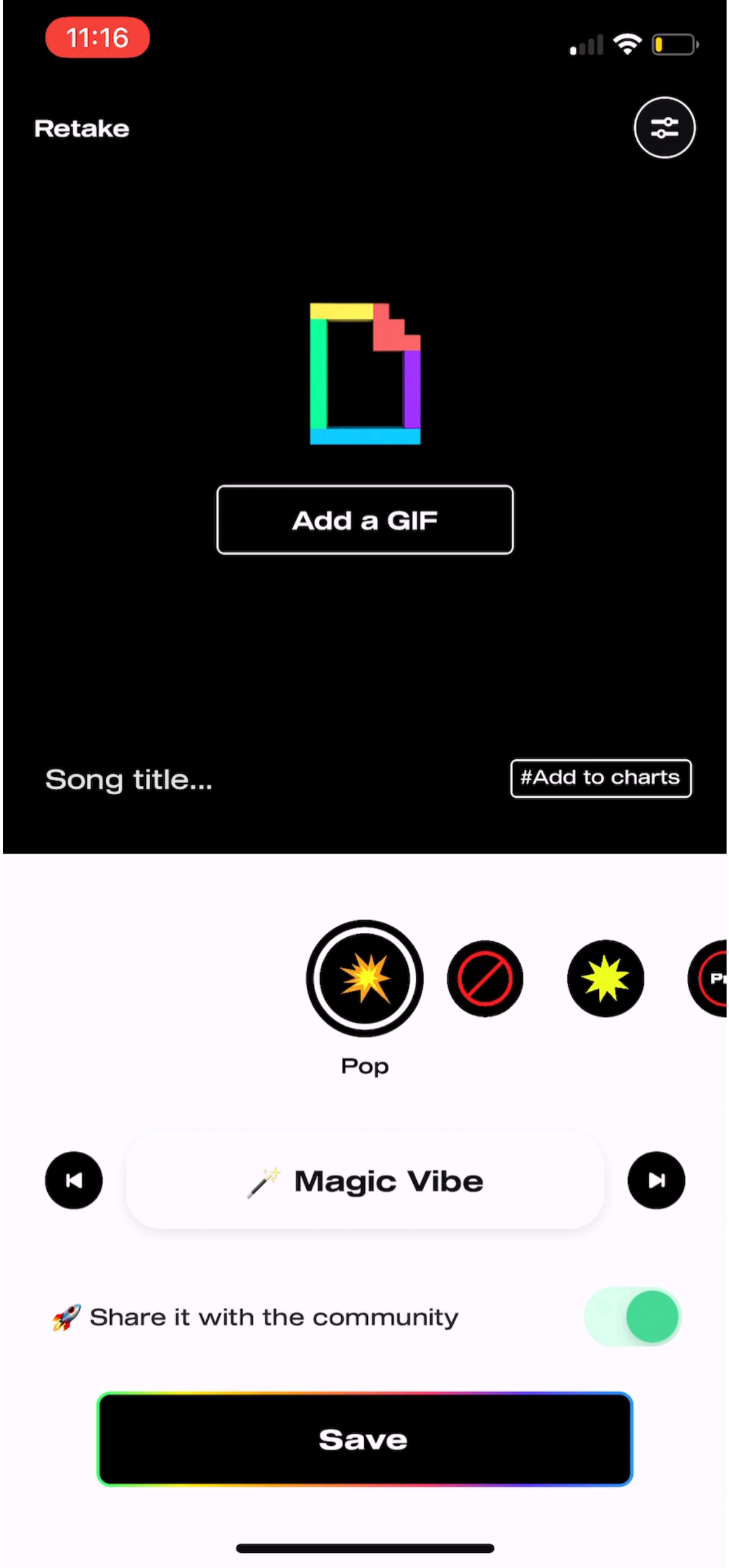
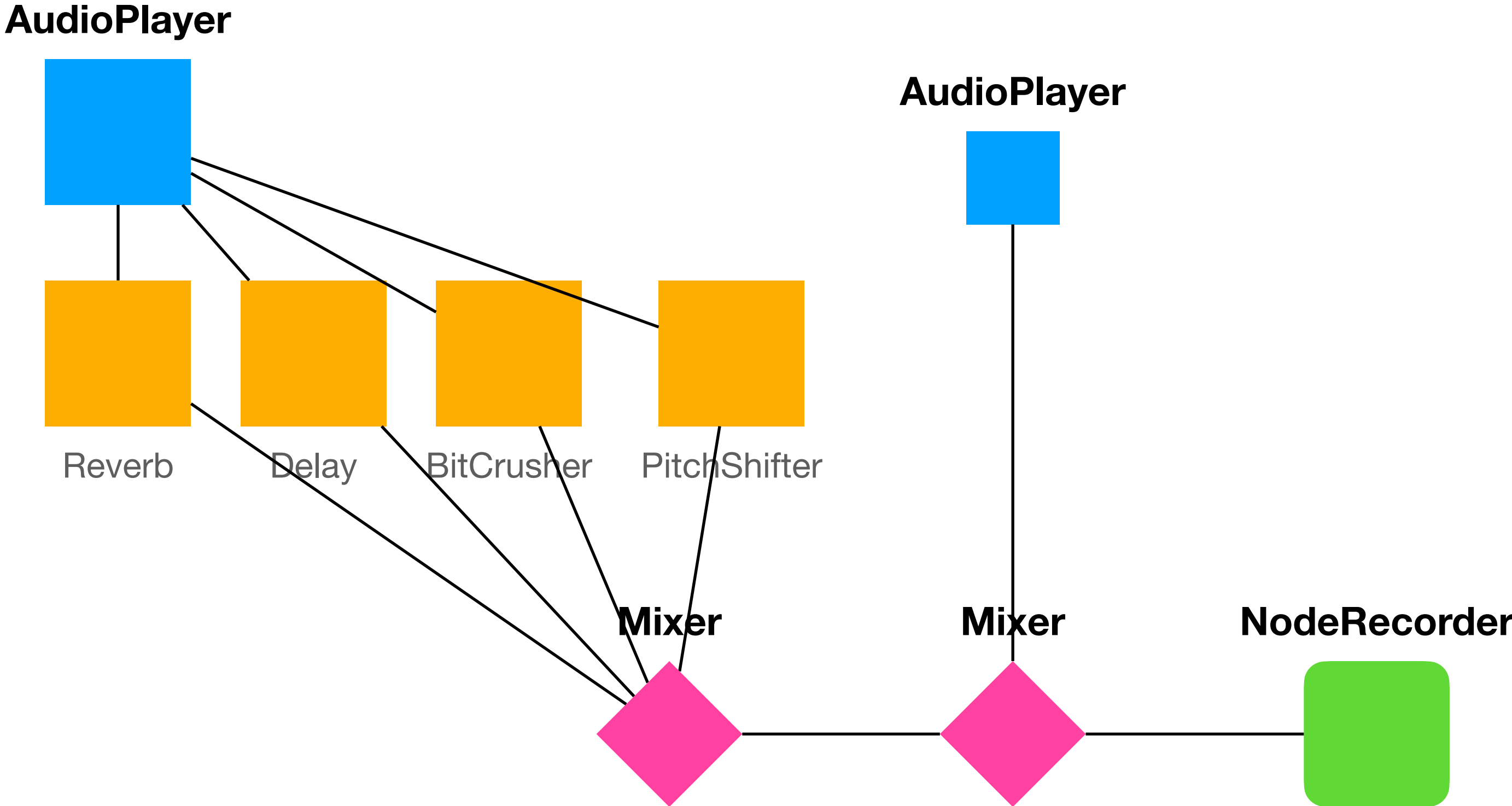


“99% of people should be using AudioKit... [it’s] a fabulous and tremendously well-supported and maintained audio framework, based upon Apple's robust new AVAudioEngine. It's feature rich, newbie-friendly and comes with a bunch of great documentation and sample code, as well as a lot of great educational resources, like Swift playgrounds. No realtime C programming necessary.” — Mike Tyson

MVP in AudioKit v4 (Recording)

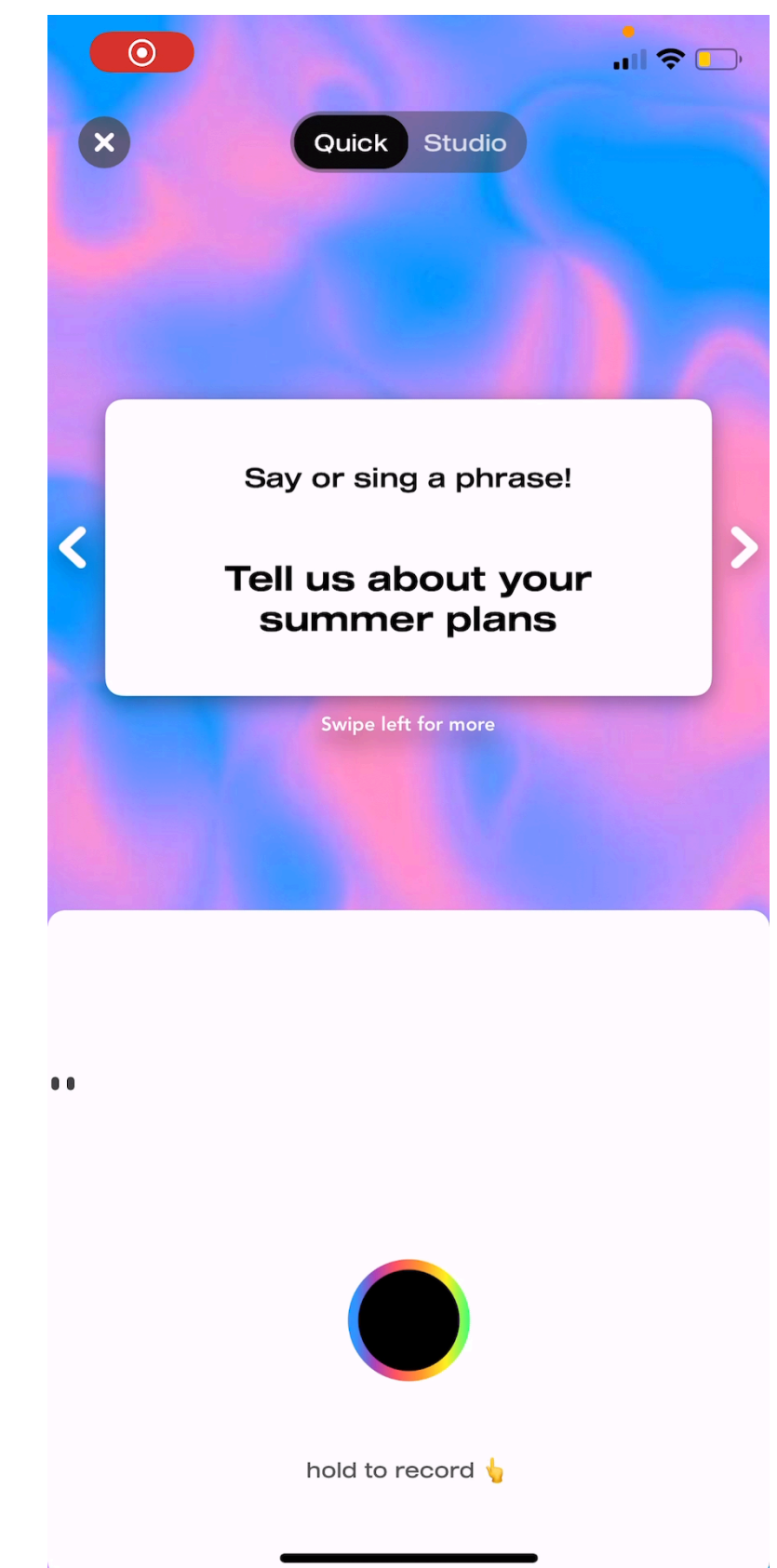
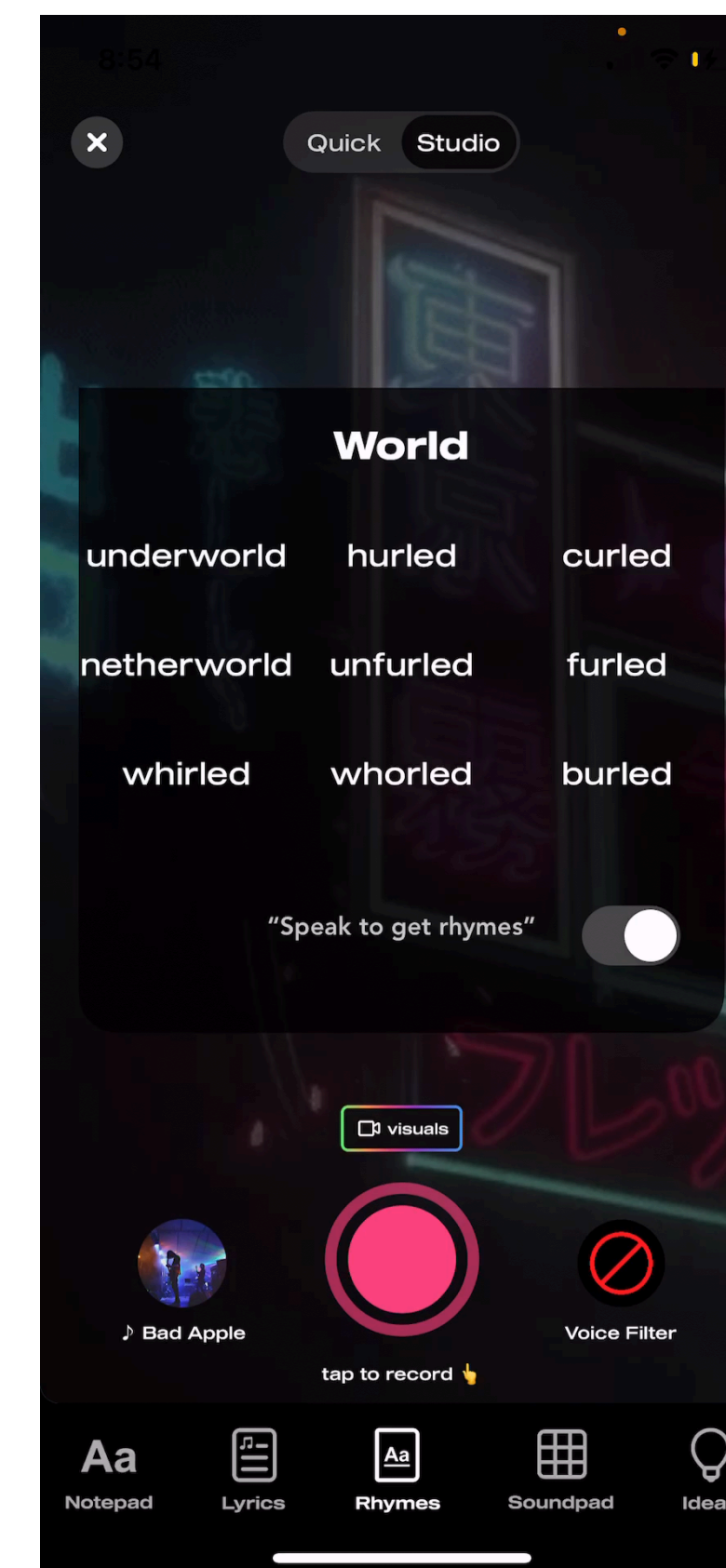
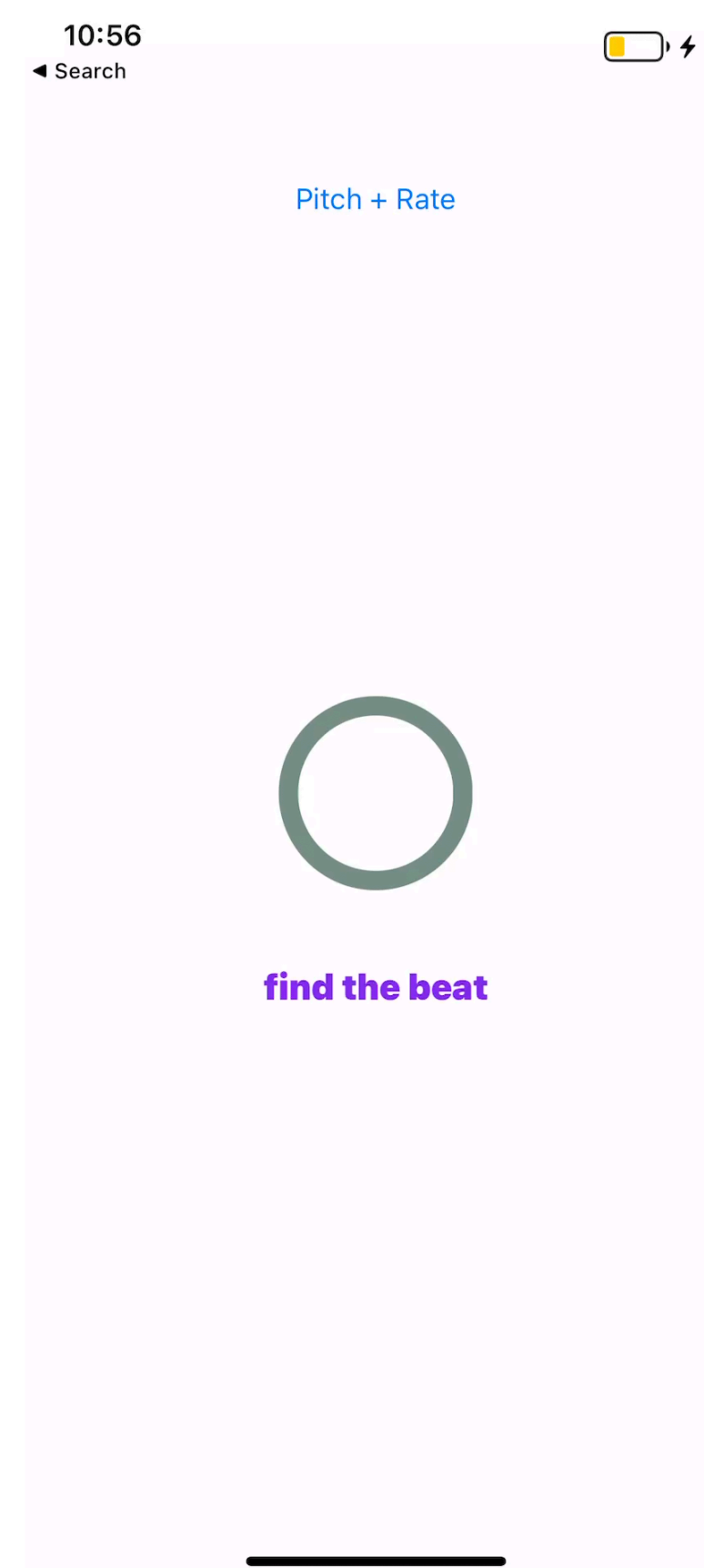


Editing (similar to Recording)



Prototyping all sorts of new audio experiences with AudioKit

- Easy to add taps, create new graphs, build new interfaces
 - Speech to Text (get rhymes by tapping input node)
- Quick create
- Modify a beat (swap out different loops)



Feedback from MVP

- Loved the experience, but wanted more from
 - **Vocal effects** — creators still wanted to sound better
 - **Ease of editing & recording**
 - We didn't have abstraction of “clips” or “tracks”

Prototyping vocal effects

- Compiling Faust
 - Modifying Soundpipe in AudioKit
 - No pitch correction (not available in Faust)
- Prototyping in Supercollider
 - Lots of easy-to-use ugens (Tartini, PitchShiftPA)
 - Need to still translate DSP back to C++

```
sig = [Mix([
  PitchShiftPA.ar(filtered, inPitch,
    midiDiff[0].midiratio, //pitchRatio
    1, //formantRatio
  ),
  // PitchShift.ar(in, 0.2, pitchRatio: f3 / inFreq),
  // PitchShift.ar(in, 0.2, pitchRatio: f4 / inFreq),
  PitchShiftPA.ar(filtered, inPitch,
    midiDiff[1].midiratio, //pitchRatio
    1, //formantRatio
  ),
  PitchShiftPA.ar(filtered, inPitch,
    midiDiff[2].midiratio, //pitchRatio
    1, //formantRatio
  ),
  PitchShiftPA.ar(filtered, inPitch,
    midiDiff[3].midiratio, //pitchRatio
    1, //formantRatio
  ),
]), LPF.ar(Mix(
  notes + [
    PitchShiftPA.ar(in, inPitch, midiDiff[0].midiratio / 2, 1),
    PitchShiftPA.ar(in, inPitch, midiDiff[0].midiratio * 2, 1),
    PitchShiftPA.ar(in, inPitch, midiDiff[0].midiratio, 1)
  ]
))]
sig
```

Building better vocal effects with DSP

~~Using AudioUnit v3~~ with AudioKit v5

Sample Code

Creating Custom Audio Effects

Add custom audio effect processing to apps like Logic Pro X and GarageBand by creating Audio Unit (AU) plug-ins.

Download



Availability

IOS 13.0+
iPadOS 13.0+
macOS 10.15+
Xcode 11.6+

Framework

Audio Toolbox

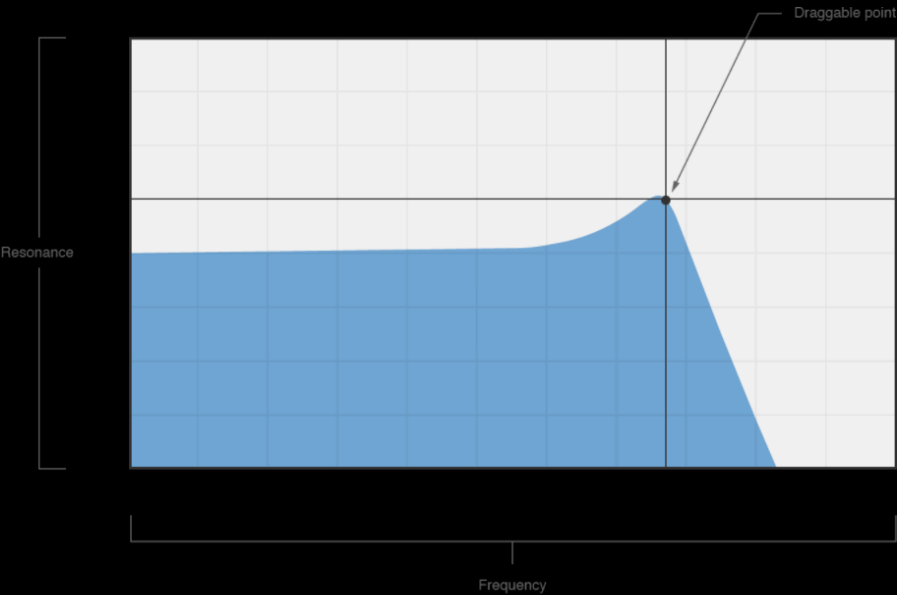
On This Page

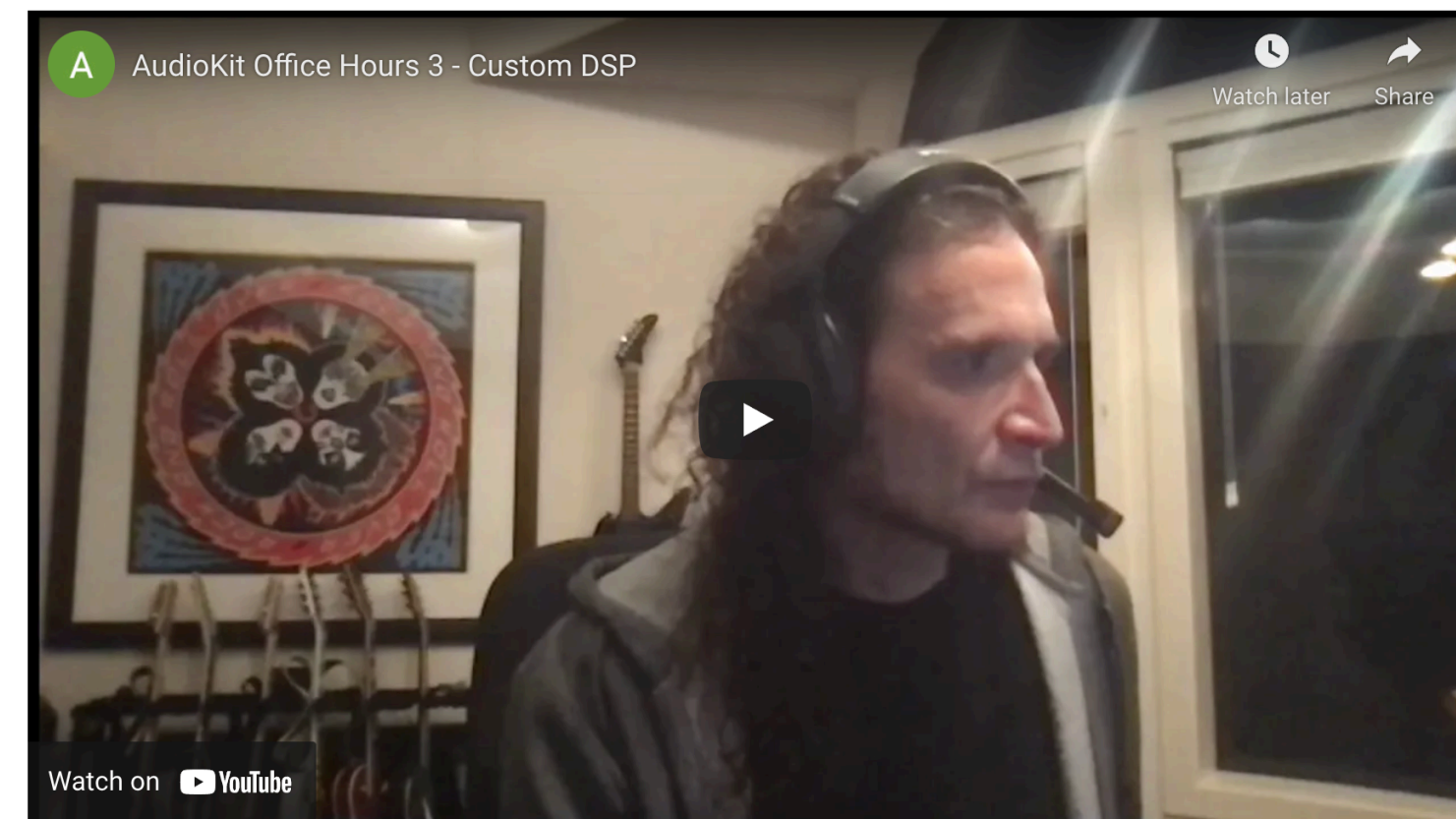
[Overview](#) 
[See Also](#) 

Overview

This sample app shows you how to create a custom audio effect plug-in using the latest Audio Unit standard (AUv3). The AUv3 standard builds on the [App Extensions](#) model, which means you deliver your plug-in as an extension that's contained in an app distributed through the App Store or your own store.

The sample Audio Unit is a low-pass filter that allows frequencies at or below the cutoff frequency to pass through to the output. It attenuates frequencies above this point. It also lets you change the filter's resonance, which boosts or attenuates a narrow band of frequencies around the cutoff point. You set these values by moving the draggable point around the plug-in's user interface as shown in the figure below.





By AudioKit Pro | [Blog](#) | [Office Hours](#) | [Tutorials](#) | January 22, 2021

1 < ≡ >

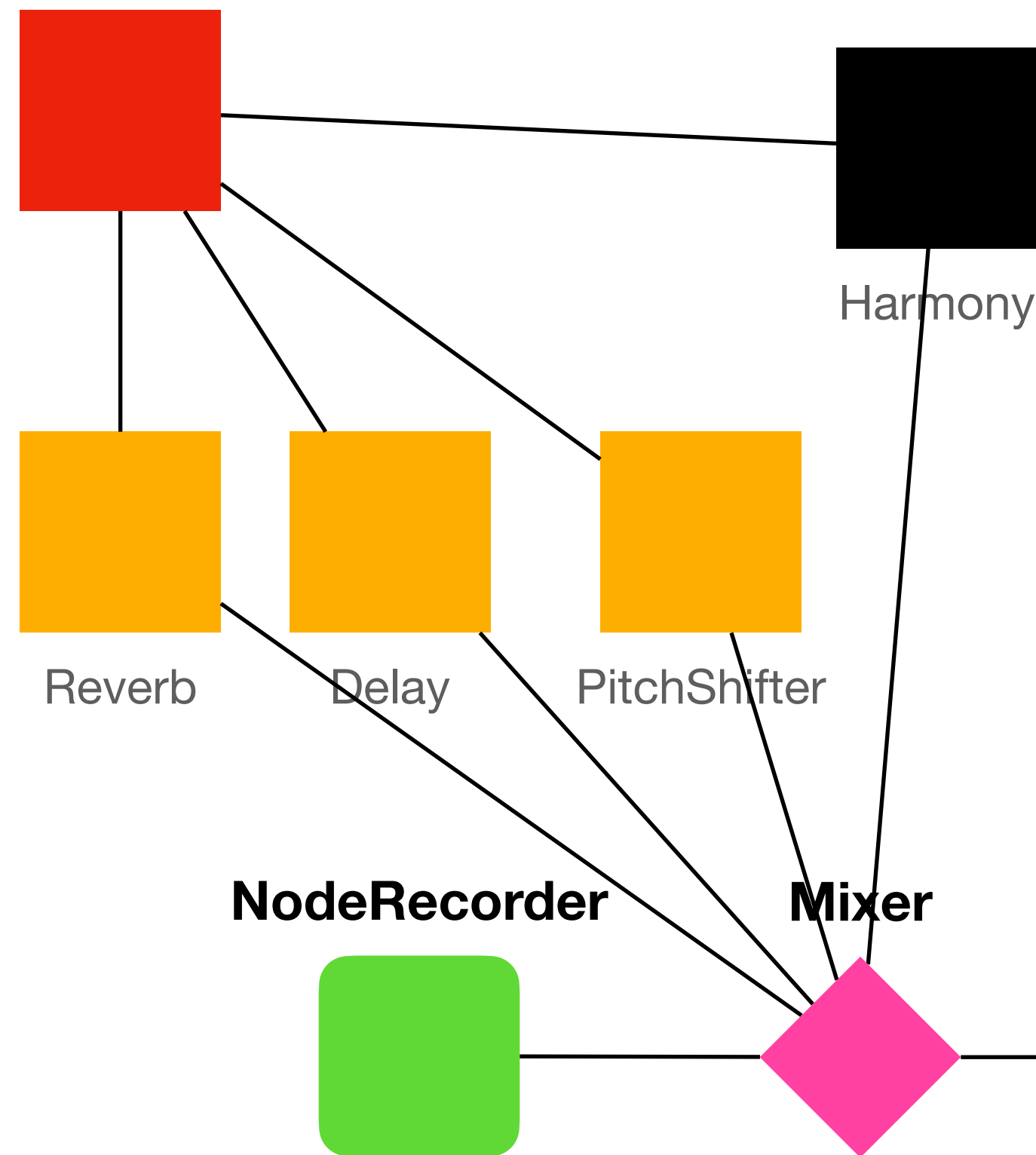
Office Hours #3 was a DSP (Digital Signal Processing) deep dive teaching how to create custom AudioKit nodes and how to dig down even deeper to create your own C-level DSP. *Video above.*

```
AK_REGISTER_DSP(VocoderDSP)
AK_REGISTER_PARAMETER(VocoderParameterShift)
AK_REGISTER_PARAMETER(VocoderParameterWindowSize)
AK_REGISTER_PARAMETER(VocoderParameterCrossfade)
```

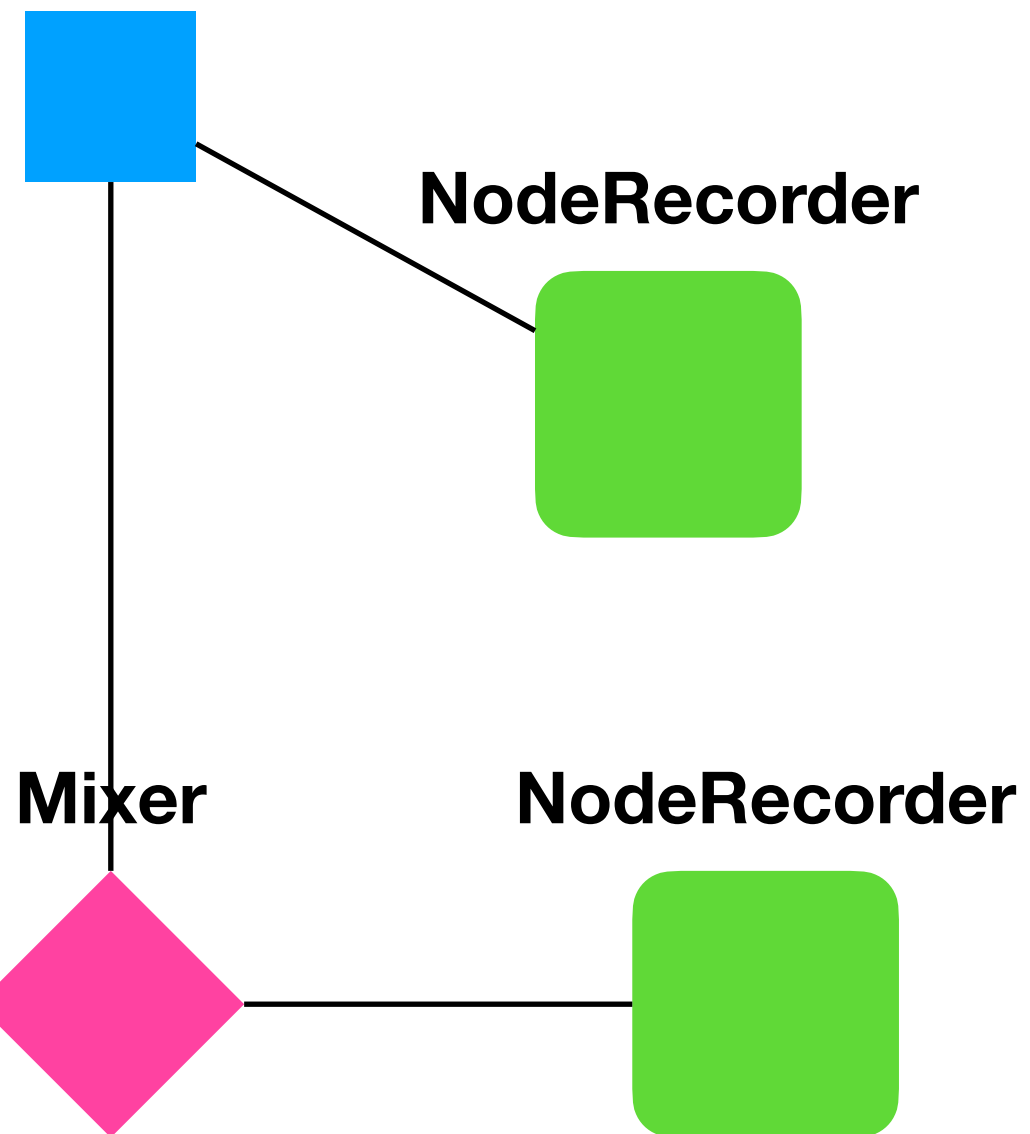
AudioKit v5

Building custom voice fx easily

AudioEngine.input



AudioPlayer



```
import AudioKit
public class Harmony: Node, AudioUnitContainer, Toggleable {
    /// Unique four-letter identifier "mkhr"
    public static let ComponentDescription = AudioComponentDescription(effect: "mkhr")

    /// Internal type of audio unit for this node
    public typealias AudioUnitType = InternalAU

    /// Internal audio unit
    public private(set) var internalAU: AudioUnitType?

    // MARK: - Parameters

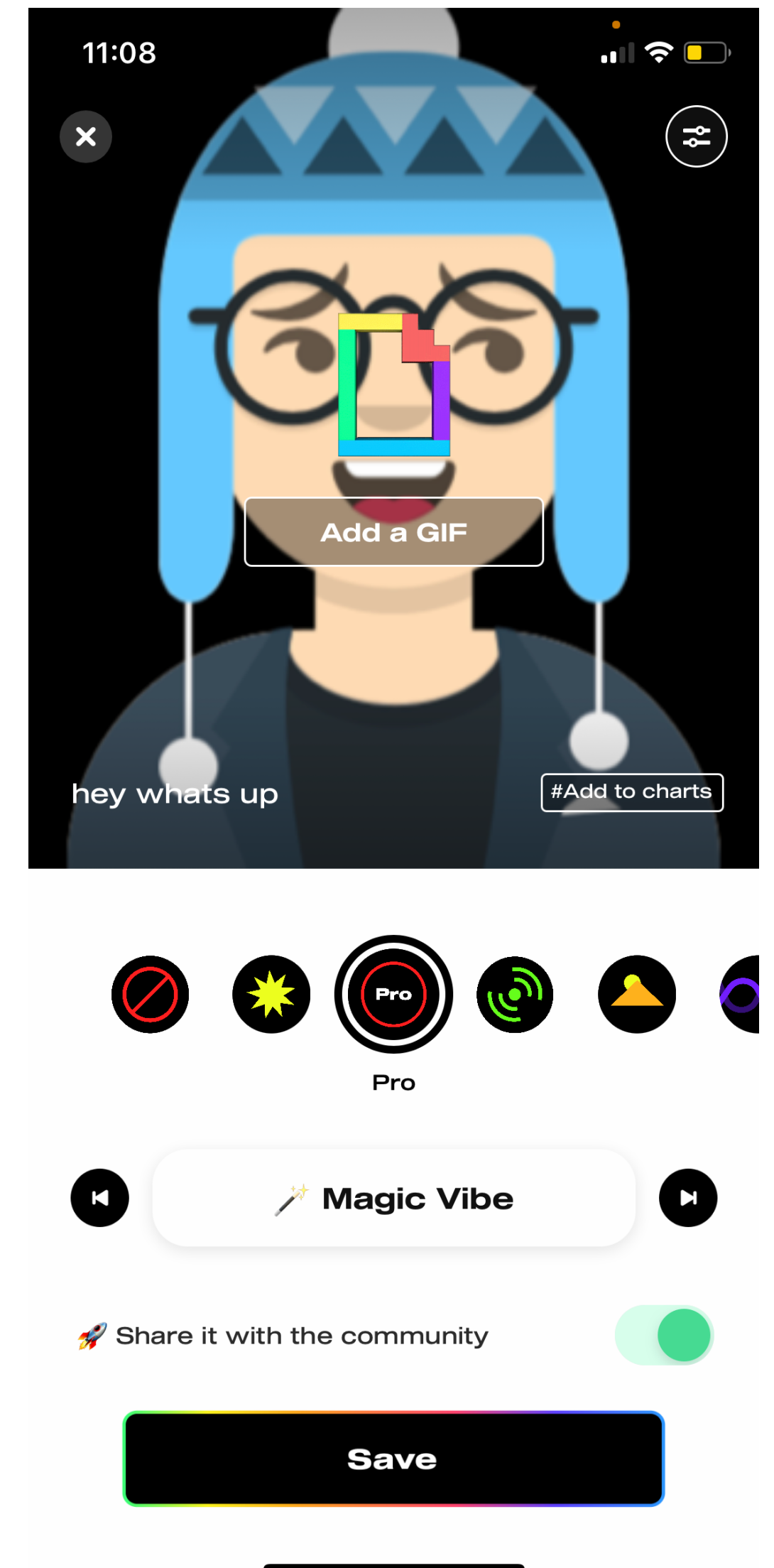
    // MARK: - Audio Unit

    /// Internal Audio Unit for Harmony
    public class InternalAU: AudioUnitBase {
        /// Get an array of the parameter definitions
        /// - Returns: Array of parameter definitions
        public override func getParameterDefs() -> [NodeParameterDef] {
            []
        }

        /// Create the DSP Reference for this node
        /// - Returns: DSP Reference
        public override func createDSP() -> DSPRef {
            akCreateDSP("HarmonyDSP")
        }
    }
}
```


Custom vocal effects

- Faust, Chuck, SOUL
- Pitch shifters
 - Rubberband
 - Elastique



Building custom voice FX with LEAF

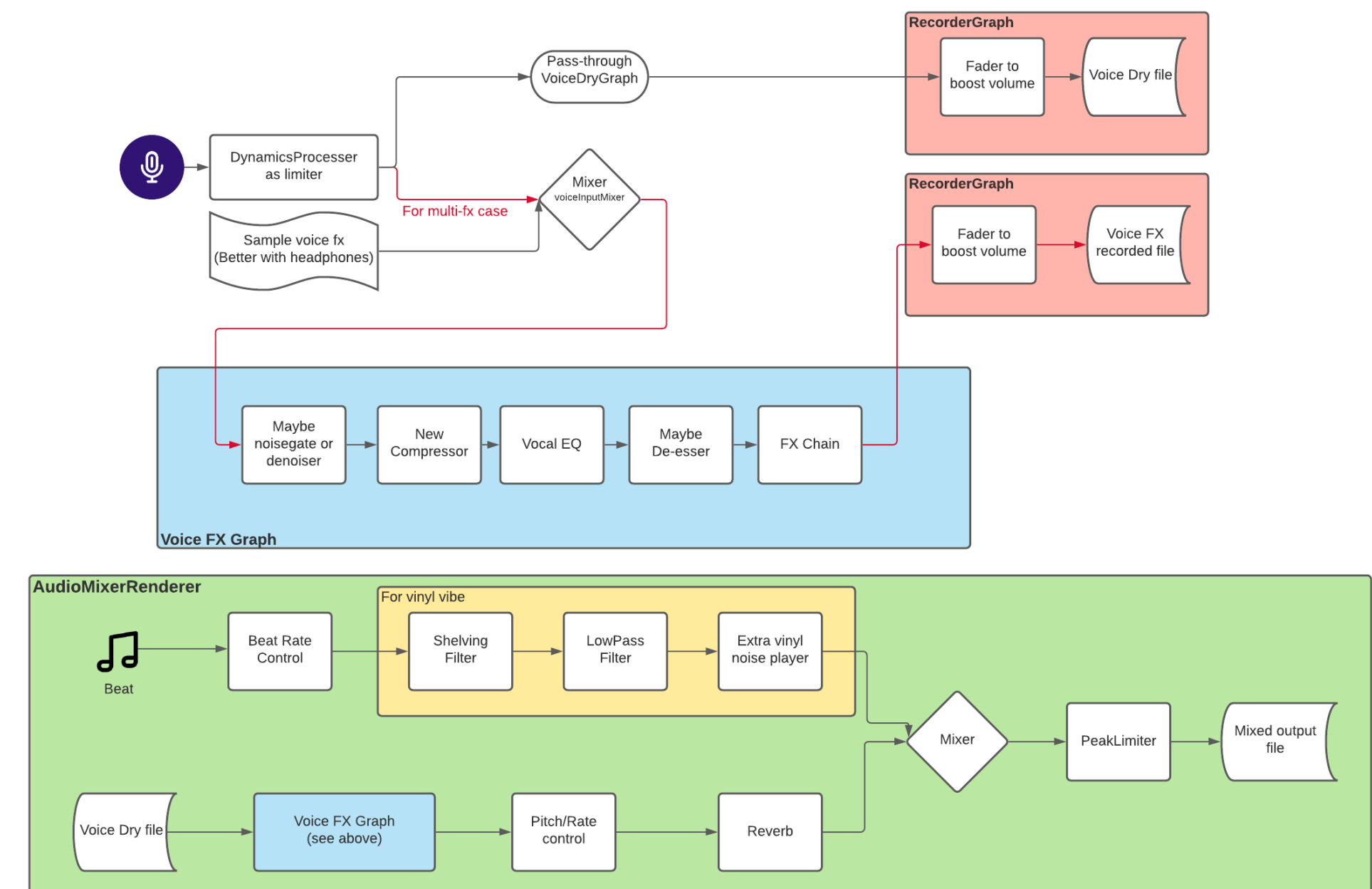
LEAF

LEAF (Lightweight Embedded Audio Framework) is a C library for audio synthesis and processing created by Jeff Snyder, Mike Mulshine, and Matt Wang at Princeton University's New Instrument Research Lab. It was originally called OOPS when we started writing it in 2017, so you may see references to it under that name as well.



Better editing tools

- Creators wanted better tools (trimming, splitting, looping)
- Tracktion offered clean abstractions (+ built atop JUCE)

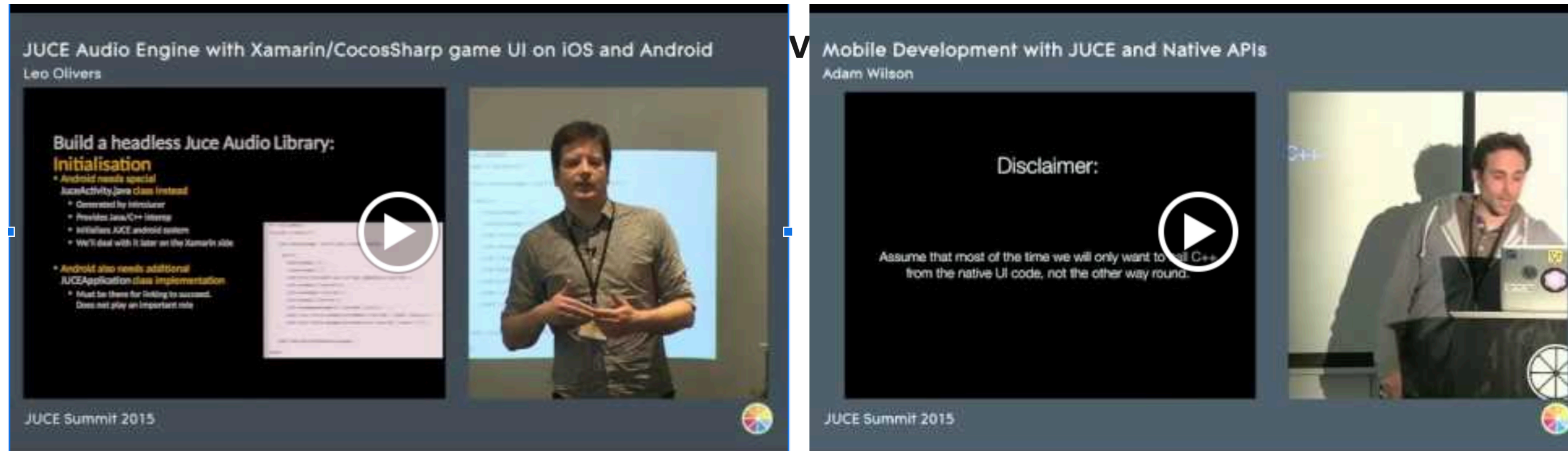


Exploring JUCE/tracktion

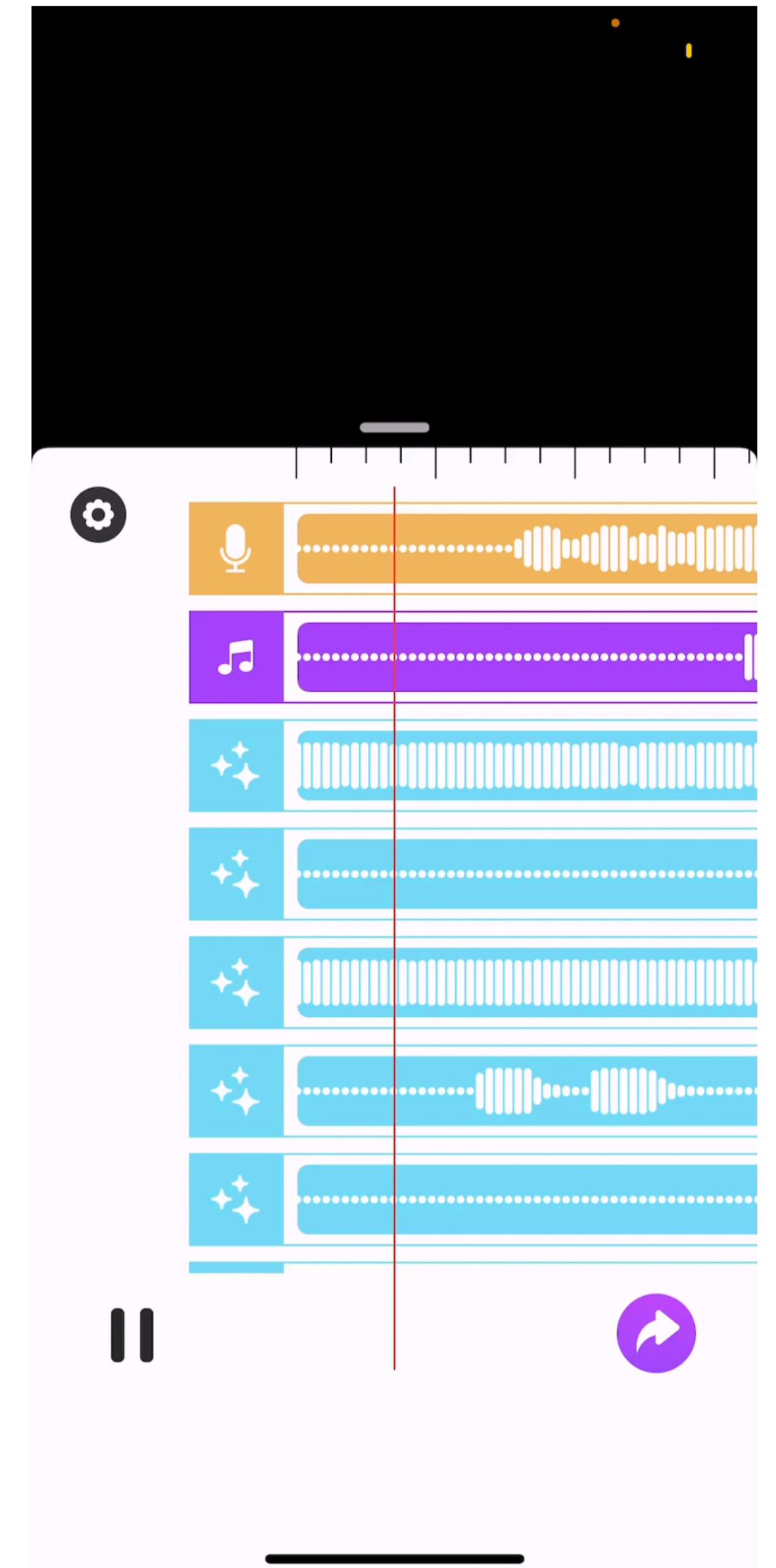
Seeking, Editing, and Loading Projects

- How to get working with native UI?

Native UI + JUCE

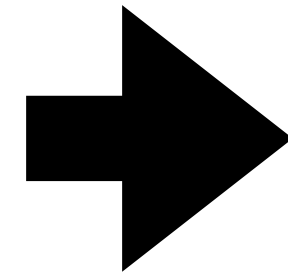


- If this feels familiar, its from 2015
- Really useful ADC talks!
- Demo up and running quickly powered by tracktion

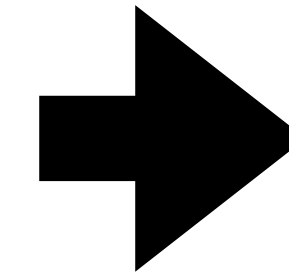



Bringing JUCE/tracktion into a native project

 CMakeLists.txt



```
▼ MaykAudioEngine
  > Source
  ▼ Bindings
    h FileSegment
    m+ FileSegment
    h MaykAudio
    h MaykAudioEngine
    m+ MaykAudioEngine
    h MaykAudioEngine_Private
  ▼ Modules
    > tracktion_engine
    > juce
    > tracktion_graph
```



 MaykAudioEngine.framework

🌟 invite

/cube

💎 7.4k



cube

edit profile

821
listens

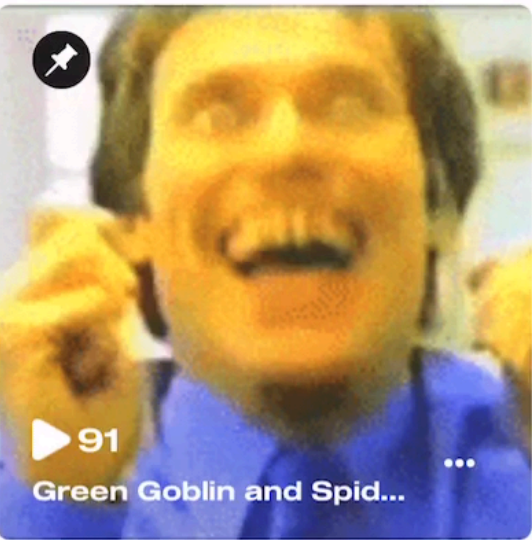
36
fans

38
following

Here to collect mayk gems not make friends

/21 Drops 168 Private Liked

Sort By



Unlocking more features

- Richer editing experiences, larger project management
- Audio development can be done outside of iOS project
- Developing new vocal FX chains in C++
- More creative recording tools (looping, layering)
- Better “remix to collab” tools

Overview

- AudioKit
- Faust
- Supercollider
- JUCE
- Tracttion Engine

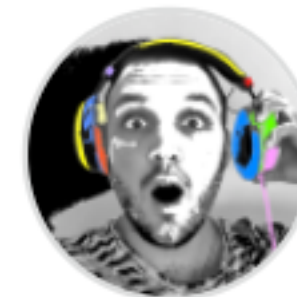
Special Thanks



Leo Olivers
leoogh



Adam Wilson
adamski



Mike Mulshine
mulshine



Jeremy Klein
jklein24



Andrei Rychkov
Moonko



Ryan Michael McGee
ryanmichaelmcgee

- Franciso Garcia
- David Gibson
- Lance Jabr
- Spencer Salazar
- Matt Howell
- Adam Nemecek
- Fabian Renn
- Aurelius Prochazka // AudioKit
- David Rowland // Tracttion
- Romain Michon // Faust
- Eyal Amir



Intrigued?

- Would love to hear from you!
 - Team Twitter: **@mayk_it_app**
 - My Twitter: **@xoob**
 - Email: **akiva@mayk.it**