

This project contains the source which builds into an iOS app. Its function is to take an input .wav file and sync the display with the music as it plays to the user.

There are two modes of operation, which the user may swipe through:

#### **Particle System Mode:**

- Particles spawn randomly in a sphere on the screen, displayed using a given texture in '.jpg' format. I used a sphere with  $\alpha = 1$  in the center and  $\alpha = 0$  at the radius
- Colors of particles represented are proportional to the frequency range at the current time interval in the song (buckets of colors separated by "instrument")
- Gravity force is in the direction of real-world gravity acting on the iOS device
- Batches of particles are destroyed and reborn when very high and very low frequencies are detected. High-frequency particles bounce upwards, while low-frequency particles are thrown downward

#### **Object View Mode:**

- An '.obj' file is loaded and displayed with a Binn-Phong shader
- The object is rotated with respect to the accelerometer of the device, and the light source is rotated with it (in the case of the sphere object file I used, this is helpful for the user to see that the object is actually moving)
- High frequencies shake the object
- Low frequencies scale the object
- This mode makes it clearer to the user how the music is being segmented for display features

## Sources:

<https://www.haroldserrano.com/blog/rendering-3d-objects-in-metal>

[https://docs.swift.org/swift-book/LanguageGuide/Generics.html#apple\\_ref/doc/uid/TP40014097-CH26-XID\\_283](https://docs.swift.org/swift-book/LanguageGuide/Generics.html#apple_ref/doc/uid/TP40014097-CH26-XID_283)

<http://metalbyexample.com/modern-metal-1/>

<http://metalkit.org/2016/02/08/using-metalkit-part-5.html>

[https://en.wikipedia.org/wiki/Rotation\\_matrix#Conversion\\_from\\_and\\_to\\_axis-angle](https://en.wikipedia.org/wiki/Rotation_matrix#Conversion_from_and_to_axis-angle)

<https://metashapes.com/blog/opengl-metal-projection-matrix-problem/>

<https://github.com/metal-by-example/sample-code/blob/master/objc/07-Mipmapping/Mipmapping/Shaders.metal>

[https://developer.apple.com/documentation/metal/hello\\_triangle](https://developer.apple.com/documentation/metal/hello_triangle)

“Camera.h” - Dr. Shinjiro Sueda, Texas A&M University

<https://github.com/k0nserv/SwiftObjLoader/blob/master/Sources/ObjLoader.swift>

“Shape.h” - Dr. Shinjiro Sueda, Texas A&M University

<https://stackoverflow.com/questions/28938660/how-to-lock-orientation-of-one-view-controller-to-portrait-mode-only-in-swift>

<https://stackoverflow.com/questions/34751294/how-can-i-generate-an-array-of-floats-from-an-audio-file-in-swift>

// FFT

[https://developer.apple.com/documentation/accelerate/vdsp/fast\\_fourier\\_transforms/finding\\_the\\_component\\_frequencies\\_in\\_a\\_composite\\_sine\\_wave](https://developer.apple.com/documentation/accelerate/vdsp/fast_fourier_transforms/finding_the_component_frequencies_in_a_composite_sine_wave)

// freq ranges

<http://www.zytrax.com/tech/audio/audio.html>

// textures

<https://www.raywenderlich.com/719-metal-tutorial-with-swift-3-part-3-adding-texture>

// alpha blending

<http://metalbyexample.com/translucency-and-transparency/>

// accelerometer

<https://www.youtube.com/watch?v=XDuchXYiWuE>

//swipe gestures

<https://github.com/calebrwells/A-Swiftly-Tilting-Planet/tree/master/2018/UIFontGestureRecognizer/Handling%20Swipe%20Gestures/Handling%20Swipe%20Gestures>

// color interpolation

<https://www.alanzucconi.com/2016/01/06/colour-interpolation/>

<https://gist.github.com/cjddmut/fefe5dac35cccfceabec>

[https://developer.apple.com/documentation/coremotion/getting\\_processed\\_device-motion\\_data](https://developer.apple.com/documentation/coremotion/getting_processed_device-motion_data)