

# Time-Driven UI Animation

by reproducing a Shazam-style animation

Aniseh Khajuei





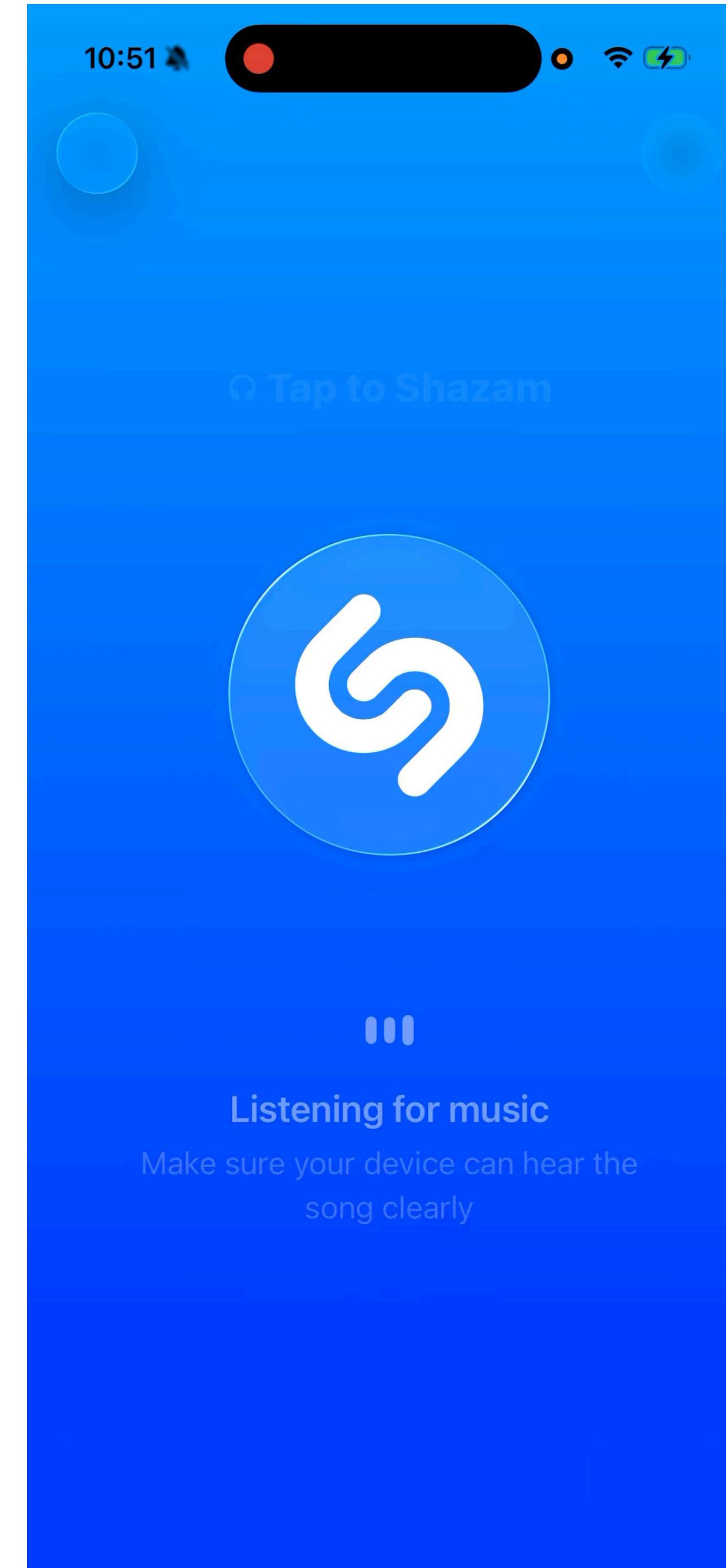
# Learning Goal

To implement and parameterize time-driven, shape based animations in SwiftUI:

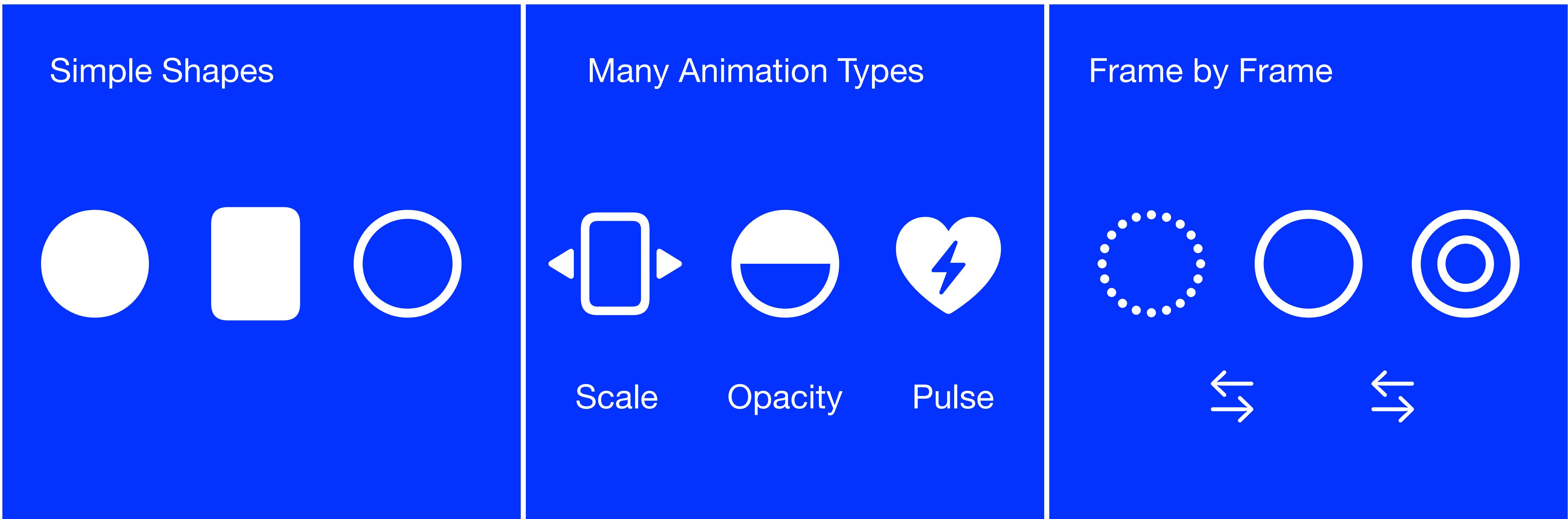
including scale, opacity, staged reveals and gated pulses.

# Case Choice

## Shazam Application



# Why SHAZAM?



Case Choice

# Implementation Approach

## How did I do it?

1. Single clock: TimelineView drives everything.
2. Fast oscillator (sin) → central breathing.
3. Slow envelope → global swell (grow → peak → shrink).
4. Phase gating (growthProgress, fmod) for staged reveals.
5. Compose final visuals =  $\text{globalScale} * \text{localOsc} * \text{pulse}$  and separate opacity logic.



Shazam main

Shazam > iPhone 17 Paused Shazam on iPhone 17 2

Pulse Animation Preview

Shazam

Assets

Bars

ContentView

PulsingRingsView

ShazamApp

ShazamPulseView

// ShazamPulseView.swift

// Shazam

//

// Created by Aniseh Khajuei on 10/11/25.

//

import SwiftUI

struct ShazamPulseView: View {

// ----- tuning (configuration parameters) -----

// Duration of a full cycle (seconds) = grow + shrink

let cycleDuration: Double = 2.5

// Fraction of the cycle allocated to growth (rest is for shrink)

let growthEndFraction: Double = 0.70

// How much the whole set can scale up (global scale)

let maxGlobalScale: CGFloat = 2.2

// Number of central "squeezes" during growth and their strength

let squeezesDuringGrowth: Int = 3

let squeezeAmount: CGFloat = 0.04

// Thresholds and fade-in width for disks (when they appear and how fast)

let diskAppearThresholds: [Double] = [0.05, 0.18, 0.35, 0.50]

let diskAppearWidth: Double = 0.18

// Central circle base size and disk base sizes

let centerBase: CGFloat = 100

let diskBaseSizes: [CGFloat] = [140, 180, 220, 260]

// Minimum and maximum opacity for disks

let diskMinOpacity: Double = 0.06

let diskMaxOpacity: Double = 0.2

// Large thin outline rings

// ringStartSize is base for the first ring; second ring uses its own base by spacing

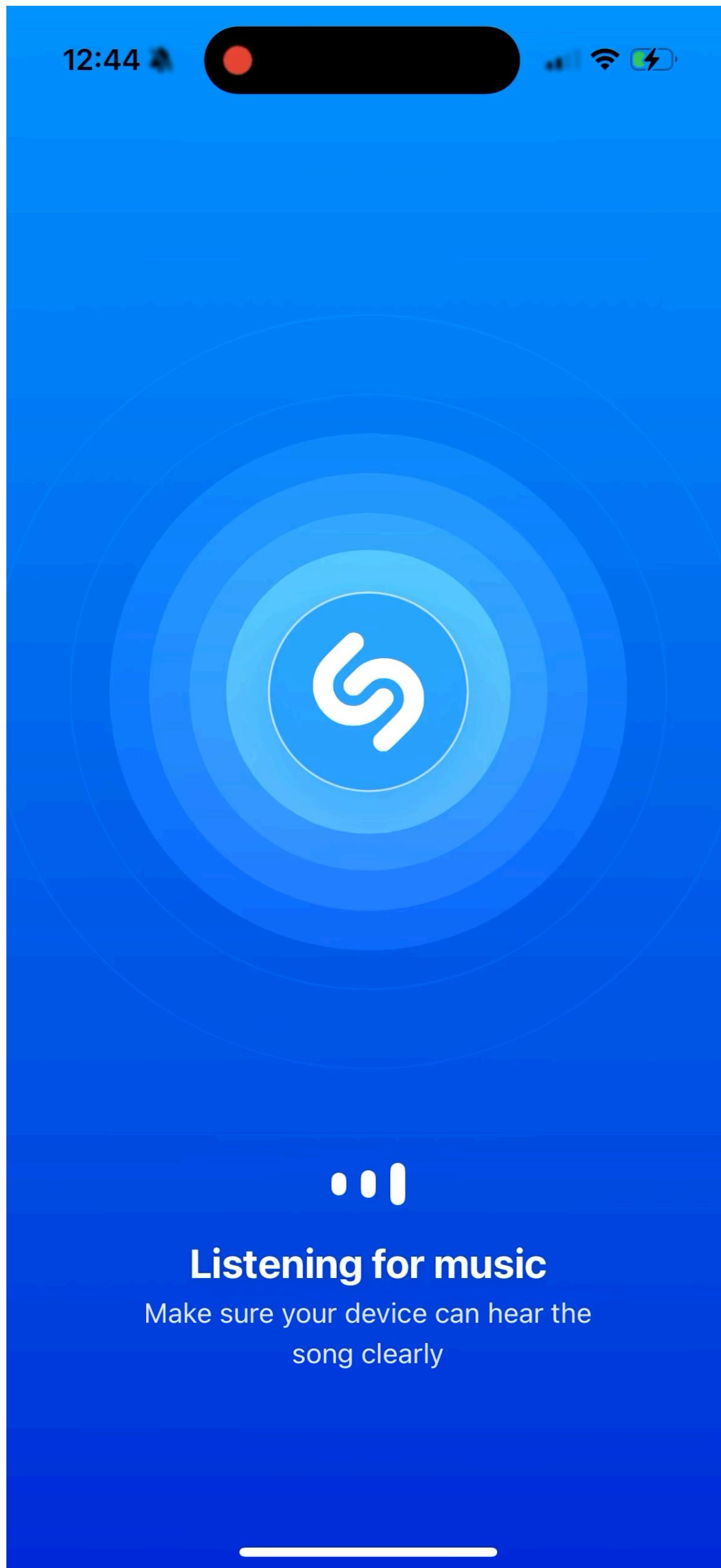
let ringStartSize: CGFloat = 300

// Fixed spacing between the two rings (kept constant)

let ringSpacing: CGFloat = 80

// Minimum/maximum opacity for rings





# Transferable Lessons



**Time-Driven  
Animation**



**Layered  
Animation**



**Phased  
Animation**



**Pulse  
Animation**



**Accessible  
Animation**

**Thank You For Your Attention**