

```

//
// ENAnimation.swift
// RGB Controller
//
// Created by Erik Nordlund on 6/15/18.
// Copyright © 2018 Erik Nordlund. All rights reserved.
//
// Arm Controller includes the following open-source components:
// • swiftBluetoothSerial: https://github.com/hoiberg/SwiftBluetoothSerial
// • peertalk-simple: https://github.com/kirankunigiri/peertalk-simple

import Foundation
import UIKit

struct ENAnimation {

    let maxFPS = 60

    func changeColorFromLeft(forLabel: UILabel, fromColor: UIColor, toColor:
        UIColor, duration: CFTimeInterval, aperture: Int, xAlignmentView:
        UIView?/*, renderCompletion: () -> Void*/) {

        /// calculating startpoint and endpoint x coordinates, comparing label
        with xAlignmentLayer
        let startX = (xAlignmentView!.frame.minX - forLabel.frame.minX) /
            (forLabel.bounds.width)
        let endX = ((xAlignmentView!.frame.minX +
            xAlignmentView!.bounds.width) - (forLabel.frame.minX +
            forLabel.bounds.width)) / forLabel.bounds.width + 1

        renderLabelGradientColorsFromLeft(fromColor: fromColor.cgColor,
            toColor: toColor.cgColor, aperture: aperture, startX: startX, endX:
            endX, bounds: forLabel.bounds, block: {gradientColors in

            let numberOfFrames = duration * Double(maxFPS)

            let frameInterval = Double(gradientColors.count) / numberOfFrames

            let frameDuration: CFTimeInterval = Double(1 / maxFPS)

            var frame: Double = 0

```

```

        animateFrame(forLabel: forLabel, withDuration: frameDuration,
            newFrame: UIColor(patternImage: gradientColors[Int(frame)]))

        frame = 1

    Timer.scheduledTimer(withTimeInterval: frameDuration, repeats:
        true, block: {timer in
            if Int(frame) >= aperture {
                forLabel.textColor = toColor

                timer.invalidate()
            } else {
                self.animateFrame(forLabel: forLabel, withDuration:
                    frameDuration, newFrame: UIColor(patternImage:
                        gradientColors[Int(frame)]))

                frame = frame + frameInterval
            }
        })
    })
}

private func animateFrame(forLabel: UILabel, withDuration: CFTimeInterval,
    newFrame: UIColor) {

    /// modify gradient layer to move toColor to the right
    let transition = CATransition()
    transition.duration = withDuration
    transition.timingFunction = CAMediaTimingFunction(name:
        kCAMediaTimingFunctionLinear)
    transition.type = kCATransitionFade
    forLabel.layer.add(transition, forKey: nil)

    forLabel.textColor = newFrame
}

private func renderLabelGradientColorsFromLeft(fromColor: CGColor,
    toColor: CGColor, aperture: Int, startX: CGFloat, endX: CGFloat, bounds:
    CGRect, block: ([UIImage]) -> Void) {
    let gradientLayer = CAGradientLayer()

    gradientLayer.bounds = bounds

    gradientLayer.colors = []

```

```

for _ in 0..

```