**RECTÁNGULO A CÍRCULO**

import SwiftUI

struct MorpfWiew: View {

@State private var animationBegin = false // estado para animar la transformación de rect. a circulo

@State private var circle = false // estado para cuando se cambie al círculo

var body: some View {

ZStack{

RoundedRectangle(cornerRadius: animationBegin ? 30 : 10) //aquí le decimos que el cornerRadios si se produce la animación sea de 30 (esto sería el radio del círculo, que lo sacammos de la mitad de la altura del rectángulo que describimos justo debajo en 60) y si no se quede en 10

.frame(width: animationBegin ? 60 : 260, height: 60) // con respecto a la anchura al pasar a círculo también tiene que cambiar, si se pasa a círculo sera de 60 igual que la altura y si no se queda en 260.

.foregroundColor(animationBegin ? Color(.systemBlue) : Color(.systemGreen)) // para que cambie el color al pasar a círculo

.overlay(

Image(systemName: "keyboard")

.font(.system(.largeTitle))

.foregroundColor(Color.white)

.scaleEffect(circle ? 0.5 : 1.0) // con esto hacemos que la imagen del teclado al pasar a círculo se reduzca a la mitad de tamaño por si no nos entra.

)

//Aquí debajo vamos a pintar un borde al rectángulo: usamos el método .trim para que no se dibuje el borde del círculo.

RoundedRectangle(cornerRadius: animationBegin ? 40 : 20)

.trim(from: 0, to: animationBegin ? 0 : 1)

.stroke(animationBegin ? Color(.systemBlue) : Color(.systemGreen), lineWidth: 10 )

.frame(width: (animationBegin ? 80 : 280), height: 80)

}.onTapGesture { // con esto hacemos la animación cada vez que tocamos la imagen

withAnimation(Animation.spring()) {

self.animationBegin.toggle()

}

//aquí debajo hacemos otra animación explícita en la cual la imagen cambia de tamaño, es decir se animaría el scaleEffect que le aplicamos a la imagen arriba. Le añadimos un repetición continua y un delay de 0.25 segundos)

withAnimation(Animation.spring().repeatForever().delay(0.3)) {

self.circle.toggle()

}

}

}

}

struct MorpfWiew\_Previews: PreviewProvider {

static var previews: some View {

MorpfWiew()

}

}