Core Graphics

Referências

https://www.raywenderlich.com/411-core-graphics-tutorial-part-1-getting-started
https://www.bignerdranch.com/blog/core-graphics-part-1-in-the-beginning/

O que é Core Graphics?

Core Graphics, conhecido como comercialmente como Quartz, é uma API relacionada com a maioria das formas 2D em iOS. Pontos, tamanhos, vetores e retângulos fazem parte do CG.

Como usar o Core Graphics?

Ele pode ser usado de forma simples, já que toda a UIView possui a função de draw.

```
func draw(_ rect: CGRect)
```

Essa função por default não faz nada, porém quando implementamos algo dentro dela, é usada para fazer desenhos e formas que desejamos para aquela view. Essa função é chamada quando a **View é criada ou invalidada**, ou seja, quando algo foi alterada nela. **Nunca** devemos chamar essa função diretamente, e sim a setNeedsDisplay()

O que podemos usar de forma mais básica para criar formas é a UIBezierPath. Ela é referente a Path que representa aquela forma que desejamos criar, logo, ela pode receber atributos como cor e forma.

A Bézier pode ser inicializada como um retângulo simples, porém também podemos usar formas geométricas mais complexas, como arcos.

Uma definição básica para ela seria:

```
let plusPath = UIBezierPath()

plusPath.lineWidth = 4

plusPath.move(to: CGPoint(x: rect.midX , y: rect.midY – plusSize))

plusPath.addLine(to: CGPoint(x: rect.midX , y: rect.midY + plusSize))

UIColor.white.setStroke()
plusPath.lineCapStyle = .round
plusPath.stroke()
```

Onde criamos uma linha simples, definindo seu começo e seu final por meio do **move**, onde posicionamos um ponto, e o **addLine**, onde traçamos a reta do ponto inicial até esse ponto final dado. Além disso, definimos que seu **setStroke(),** ou seja, sua borda, será branca, e adicionamos o **stroke()** na Path.

Core Animation

Referências

https://developer.apple.com/documentation/quartzcore
https://jamesonquave.com/blog/core-animation-swift-tutorial-animatable-properties/

O que é Core Animation?

É um meio de criar animações, assim como criamos objetos com o Core Graphics. Core Animation também usa o Quartz para performance, de modo que a maior parte de desenhar os frames da animação é feito por ele, e temos que apenas colocar parâmetros como começo e final da animação. Porém, não é somente animar os objetos em tela que o CA é capaz de fazer, pois também **podemos manipular qualquer conteúdo visual.**

Como usar o Core Animation?

Podemos usar de diversas formas, inclusive, muitos já devem ter usado sem saber que se tratava dele.
A principal parte do CA é o CAlayer. Toda UIView que criamos possui uma, porém ela pode existir sem o componente de view. Ela possui atributos como background color, border e shadow. Essas duas ultimas costumamos usar quando queremos colocar borda arredondada ou sombra na view. Podemos colocar imagens também, se for essa a ideia.

```
let redLayer = CALayer()

redLayer.frame = CGRect(x: 50, y: 50, width: 50, height: 50)
redLayer.backgroundColor = UIColor.redColor().CGColor

 // Round corners
 redLayer.cornerRadius = 25

 // Set border
 redLayer.borderColor = UIColor.blackColor().CGColor
 redLayer.borderWidth = 10

 // Set Shadow
 redLayer.shadowColor = UIColor.blackColor().CGColor
 redLayer.shadowOpacity = 0.8
```

```
redLayer.shadowOffset = CGSizeMake(2, 2)
redLayer.shadowRadius = 3
```

Podemos criar também animações com o CA, usando o CABasicAnimation, que recebe como atributo um **keyPath**, que é o "nome" que desejamos dar para essa animação. Exemplos de nomes seria "position", "transform.scale" entre outros.

```
// Create a blank animation using the keyPath "cornerRadius", the property we want to animate
let animation = CABasicAnimation(keyPath: "cornerRadius")

// Set the starting value
animation.fromValue = redLayer.cornerRadius

// Set the completion value
animation.toValue = 0

// How may times should the animation repeat?
animation.repeatCount = 1000

// Finally, add the animation to the layer
redLayer.addAnimation(animation, forKey: "cornerRadius")
```

Podemos fazer efeitos como esse do lado, criando as barras como CAShapeLayer e anima-los.

Detalhes importantes: Para formas básicas, como retângulos, o uso de CALayers se torna mais interessante, pois podemos colocar o Path direto nela. Por outro lado, quando tentamos fazer algo mais complexo, podemos criar um UIBezierPath() e colocar ele dentro do CALayer.

Animações mais complexas podem ser feitas combinando as mudanças de Layer junto com as animações de UIView().animate.

Existem diversas formas de animar, e as principais funções são:

- CABasicAnimation
- CAKeyframeAnimation
- CAAnimationGroup
- CATransition