

# iOS中的绘图概念 - By Dorayo

## 1. iOS中的绘图概念

iOS 提供了两种主要方式来创建高质量的图形：OpenGL或者使用Quartz、Core Animation和UIKit进行原生渲染。

Quartz 是最主要的绘图接口，提供了基于路径的绘制和反走样渲染、渐变绘制、坐标系转换和PDF文档的创建和展示。

UIKit 提供了线条、图片和颜色操作的Objective-C接口的封装。

Core Animation 为很多UIKit的view的属性动画提供的底层支持，并且也可以用来实现一些自定义的动画。

### 1.1 UIKit 图形系统

在iOS中，所有要往屏幕上绘制的内容都需要UIView对象来实现，而不管到底是使用OpenGL、Quartz、UIKit 还是 Core Animation。

如果使用系统预先提供的UIView子类，绘图的处理会自动进行，我们不需要关心细节。

如果要自定义视图，我们必须自己实现绘图代码。如果我们使用 Quartz、Core Animation 或者UIKit 来绘制，就必须先了解如下绘图概念

#### 1.1.1 视图的绘制周期

当一个视图第一次显示或者视图的某些部分需要重绘，iOS系统会向该视图对象发送drawRect: 消息

有如下几种情况会触发视图进行更新：

- 移动或者移除视图对象
- 将已经隐藏的视图对象的hidden值改为NO
- 将屏幕外面的内容滚动回屏幕
- 显式地调用视图的setNeedsDisplay 或者 setNeedsDisplayInRect: 方法

注意：永远不要自己调用drawRect:方法，系统会在合适的时候去调用它

#### 1.1.2 iOS绘图中的坐标系统



在调用视图的drawRect:方法之前，UIKit通过会通过创建一个图形上下文来建立默认的坐标系，这个默认的坐标系统是ULO(upper-left-origin coordinate system)

另外，OS X使用的绘图默认坐标系统是LLO (lower-left-origin coordinate system)

### 1.1.3 点 V.S. 像素

One point does not necessarily correspond to one physical pixel

### 1.1.4 获取图形上下文

#### 1. 绘制到屏幕

调用UIGraphicsGetCurrentContext函数来获取图形上下文

#### 2. 绘制到位图上下文或PDF上下文

Drawing and Creating Images   Generating PDF Content

### 1.1.5 颜色和颜色空间

Quartz中可以通过CGContextSetRGBStrokeColor 和 CGContextSetRGBFillColor 来创建和设置颜色

## 1.2 使用Quartz和UIKit绘图

Quartz是iOS中原生绘图技术的通用名字，Core Graphics框架是其核心，并且是我们绘图的主

要接口。这个框架提供了如下数据类型和功能：

- Graphics contexts
- Paths
- Images and bitmaps
- Transparency layers
- Colors, pattern colors, and color spaces
- Gradients and shadings
- Fonts
- PDF content

UIKit实现了Quartz中最常用的一些图形相关操作，包括如下的类和功能：

- UIImage, which implements an immutable class for displaying images
- UIColor, which provides basic support for device colors
- UIFont, which provides font information for classes that need it
- UIScreen, which provides basic information about the screen
- UIBezierPath, which enables your app to draw lines, arcs, ovals, and other shapes.
- Functions for generating a JPEG or PNG representation of a UIImage object
- Functions for drawing to a bitmap graphics context
- Functions for generating PDF data by drawing to a PDF graphics context
- Functions for drawing rectangles and clipping the drawing area
- Functions for changing and getting the current graphics context

#### 1.2.1 配置图形上下文

#### 1.2.2 创建和绘制Path

#### 1.2.3 创建样式(patterns)、渐变和阴影

#### 1.2.4 自定义坐标系

### 1.3 应用Core Animation的动画效果