

# Quartz2D 学习

## 章节二 <Quartz 2D实战>

### 实战一

“

实战1，你将通过使用第一章的知识，结合Apple提供的 [Sample Code](#)，绘制一个卡通人物。效果如下图

Carrier 1:10 AM



通人物。效果如下图

那么如何实现该效果呢。

首先我们想要绘制的图像，是没有给出具体的坐标的，需要我们依靠想象在图层中绘制，也就是说所有Quartz 2D的函数中使用的坐标点需要我们自己给出，为了可以快速找到坐标位置，我们可以添加一下方法帮助我们找到坐标点：

```
-(void)touchesEnded:(NSSet *)touches withEvent:(UIEvent *)event{  
  
    //输出点击时的坐标点  
    NSLog(@"%@",NSStringFromCGPoint([[touches anyObject]  
    locationInView:self]));  
  
}
```

然后我们就可以在函数中快速的添加参数了。

### Step1

## 基本的准备工作

1. 创建一个 `CartoonView`，父类是 `UIView` 即可
2. 重写 `-(void)drawRect:(CGRect)rect` 方法
3. 添加前文提到的 `touches` 方法
4. 定义方法 `-(void)drawDoraemon:(CGContextRef)context`，来绘制哆啦A梦
5. 调用该方法

```
- (void)drawRect:(CGRect)rect {  
  
    [self drawDoraemon:UIGraphicsGetCurrentContext()];  
  
}
```

## Step2

把绘制工作拆分成以下几步：

1. 外部轮廓
2. 内部轮廓
3. 眼睛
4. 鼻子
5. 嘴巴
6. 胡须

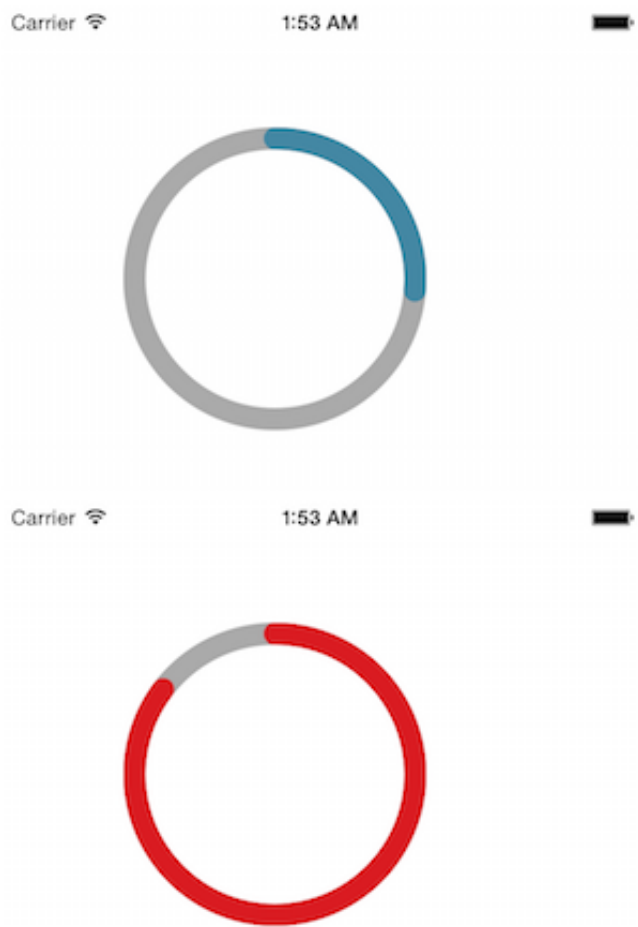
## Step3

根据绘制的需要分析出更具体的步骤，需要使用的函数 详细步骤请参考代码示例

## 实战二

“

实战2，通过实战一你应该已经熟练掌握了 `Quartz 2D` 绘制的逻辑。函数的使用需要更漫长的过程不断加强，看到不会使用的函数并不需要慌张，你需要做的仅仅是去Google一下或者查看一下文档即可（更推荐查看文档方式）。与实战一不同的是，实战二中将会结合更实际的需求最终完成一个控件，需要更加注重代码逻辑的分析和推理，我们来看一下最终效果



## 需求分析

1. 该控件最终作用如同 `UIProgressView` 一样，可以提示进度，或者倒计时等功能。
2. 该控件应该有最大值 `maxValue`，最小值 `minValue`，当前值 `value`
3. 该控件应具有设置 `value` 时，可以自动带有过渡动画功能
4. 颜色可以根据设置初始值和最终值，并且根据 `value` 值过渡

## 实现过程

### Step1

创建类 `CNProgressView`，基于 `UIView`

### Step2

声明基本属性：

- `value`
- `maxValue`
- `minValue`

声明方法:

- `setValue:animated`

### Step3

重写 `-(void)drawRect:(CGRect)rect` 方法

根据value和maxValue 绘制圆

详细步骤请参考代码示例