

# 支持点击交互的Lottie-iOS篇

快手电商无线团队 LV.5

快手电商

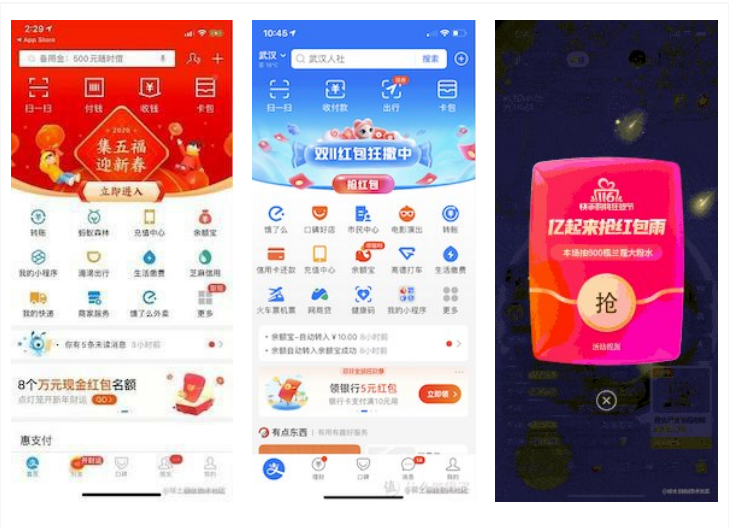
2021年12月22日 12:23 · 阅读 14467

## 0x0 背景

Hernan Torrisi在2015年创建出Bodymovin插件，让AE能够导出JSON描述的动画。2017年Airbnb工程师编写了可以渲染这个JSON文件的iOS和Android库，从此Lottie成为了实现复杂动画的首选方案，风靡移动客户端。



一般来说，动画的元素和样式由Lottie描述文件唯一确定，一旦产出交付便不能更改。常见的Lottie使用场景中，画布组成元素固定，部分元素有动画效果，例如支付宝在新春红包、双十一的首页氛围动图。



营销活动中，极为常见的就是炫目的红包弹窗，漂亮的动画对点击率和核销率有明显的提升效果。例如上图示例中，多元素的位移、缩放、炫光、回弹等复杂的综合动画效果，非常精美流畅。这样复杂的动画，用UIKit或CoreAnimation直接实现的沟通、编码成本非常巨大，动效设计师使用AE交付Lottie几乎是唯一的选择。

我们首先遇到的问题，就是展示数据由服务端下发，不同用户展现的内容不一样。为实现该场景下的动画需求，本文基于Lottie做了一些探索，希望给读者一些借鉴。

## 0x1 方案探索

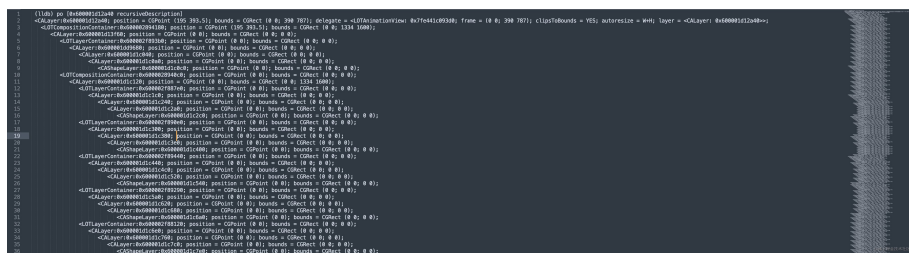
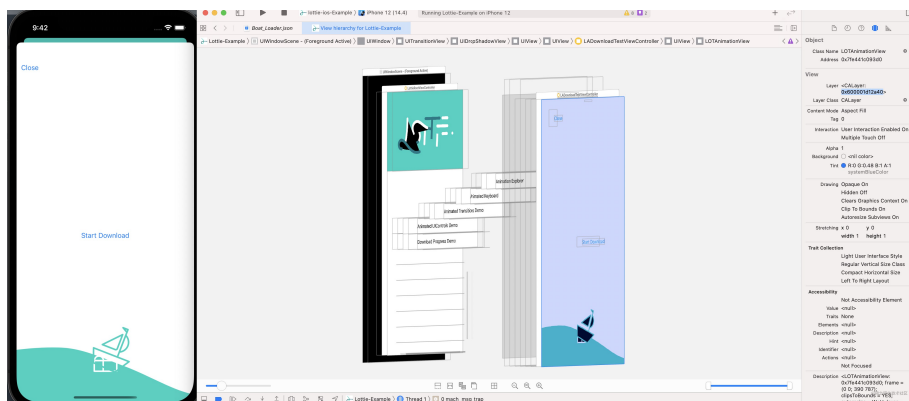
- 1. 固定元素用动画，差异元素用原生，硬凑组合
  - 缺点：何时应该显示原生控件，以及如何让原生控件贴合Lottie动画其它元素的行为轨迹，计算和调试要花费大量的时间。
- 2. Lottie文件对差异数据预留占位，服务端数据返回后替换文件中的占位字符串
  - 缺点：需要替换的文本内容比较多，部分需要替换的是图片，增加了Lottie出稿的设计成本和替换理解成本
- 3. 同层混合渲染
  - 前两种方案中，第一种方案我们在生产使用过并且深感不便，第二种方案评估后就放弃了。对于同层混合渲染的想法，一方面受到了小程序的启示，小程序将UI控件跟Webkit的DOM元素都能良好的混合，本身就是Native实现的Lottie理论上能跟UI控件更好的协同工作。

## 0x2 同层界面渲染

### 1.前置知识

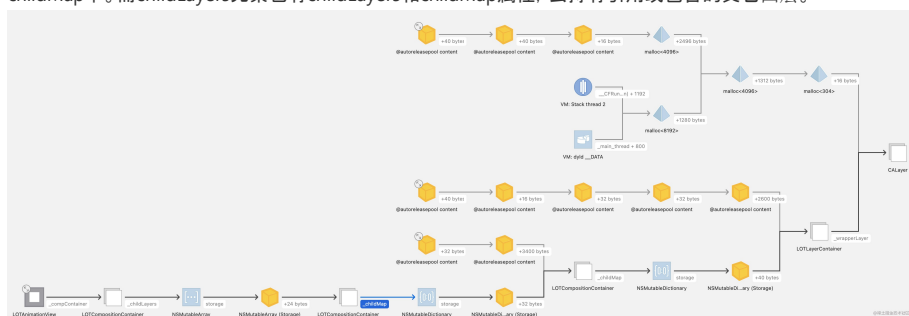
- Lottie动画控件的视图结构

Lottie动画将所有元素打平后，绘制在同一个UIView子类中，这个UIView没有任何subviews。但CALayer层级非常丰富，例如下图中的小船加载动画共包含有271个CALayer。

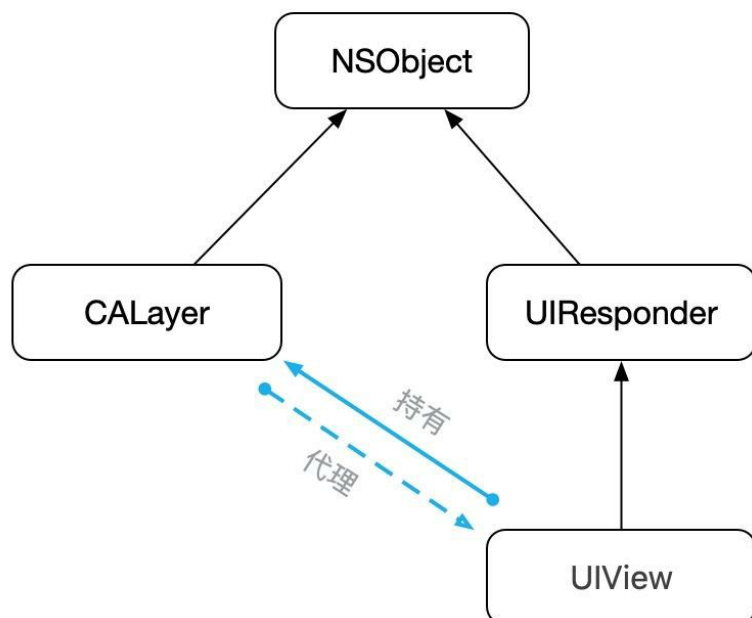


- LottieSDK对JSON图层的转换策略和图层持有关系

AE的图层包括舞台中的普通图层和不在舞台中的预合成图层，客户端解析后一一对应iOS的CALayer类实例。LottieView的compContainer作为根Layer，直接或间接持有其它所有图层，一级子图层存在属性childLayers和childMap中。而childLayers元素也有childLayers和childMap属性，去持有引用或包含的其它图层。



- iOS视图控件的组成



iOS原生框架UIKit中，所有的视图控件都继承自UIView类，UIView由CALayer+UIResponder触摸响应组成。当不需要交互时，CALayer能完成所有UIView能实现的视觉呈现。UIView持有CALayer，CALayer的delegate指向UIView。

## 2.具体实现

如前所述，CALayer负责UIView中的展示部分。直接将自定义绘制视图的CALayer，如customedView.layer，使用系统方法-[CALayer addSublayer:]添加到指定要替换的图像所在Layer，即可完成展示内容的混合。

- LottieSDK能力

[airbnb.io/lottie/#/ios...](https://airbnb.io/lottie/#/ios...)

Adding Subviews

```
AnimationView.addSubview(_ subview: AnimationSubview, forLayerAt keypath: AnimationKeypath)
```

Searches for the nearest child layer to the first Keypath and adds the subview to that layer. The subview will move and animate with the child layer. Furthermore the subview will be in the child layers coordinate space. ==Note==: if no layer is found for the keypath, then nothing happens.

Parameters : **subview**: The subview to add to the found animation layer. : **keypath**: [The keypath used to find the animation layer](#).

Example:

```
/// A keypath that finds 'Layer 1'
let layerKeypath = AnimationKeypath(keypath: "Layer 1")

/// Wrap the custom view in an 'AnimationSubview'
let subview = AnimationSubview()
subview.addSubview(customView)

/// Set the provider on the animationView.
animationView.addSubview(subview, forLayerAt: layerKeypath)
```

© 腾讯开源社区

存在的问题：

1.由于SDK添加自定义View的方法中，“.”用于表示搜索深度，如“Layer.Shape Group.Stroke 1.Color”，因此搜索图层时，图层名中的“.”后部分被忽略，即图层命名不能使用“卡.png”

2.搜索路径只支持舞台中的普通图层，不能搜索到预合成图层，git issue没有修复计划：

[github.com/airbnb/lott...](https://github.com/airbnb/lott...)

- 同层能力扩展

鉴于SDK能力支持的缺陷，需要扩展开发增加添加自定义View的方法接口，图层添加逻辑与SDK方法相同，主要更改目标图层搜索方法。

如下代码所示，先用LottieSDK方法搜索一次目标图层，如果没有搜索到，再递归遍历Lottie根图层持有的子图层树(包括预合成图层)。

ini 复制代码

```
- (void)addSubview:(nonnull UIView *)view
withLayerName:(nonnull NSString *)layerName {
    CALayer *layer = [self searchWithLayerName:layerName];
    if (layer) {
        [self _layoutAndForceUpdate];
        CGRect viewRect = view.frame;
        LOTView *wrapperView = [[LOTView alloc] initWithFrame:viewRect];
        view.frame = view.bounds;
        view.autoresizingMask = UIViewAutoresizingFlexibleWidth | UIViewAutoresizingFlexibleHeight;
        [wrapperView addSubview:view];
        [self addSubview:wrapperView];
        layer.contents = nil;
        [layer addSublayer:wrapperView.layer];
        [self.mixLayers addObject:view.layer];
    }
}

- (CALayer *)searchWithLayerName:(NSString *)layerName
{
    CALayer *layer = [self normalSearchWithLayerName:layerName];
    if (!layer) {
        layer = [self customSearchWithLayerName:layerName];
    }
    return layer;
}

- (CALayer *)normalSearchWithLayerName:(NSString *)layerName
{
    LOTCompositionContainer *_compContainer = [self valueForKey:@"_compContainer"];
```

```
LOTKeyPath *keypath = [LOTKeyPath keypathWithString:layerName];
CALayer *layer = [_compContainer _layerForKeypath:keypath];
return layer;
}

- (CALayer *)customSearchWithLayerName:(NSString *)layerName
{
    LOTCompositionContainer *_compContainer = [self valueForKey:@"_compContainer"];
    LOTLayerContainer *layerContainer = [self traversLayer:_compContainer withLayerName:layerName];
    return layerContainer.wrapperLayer;
}

- (LOTLayerContainer *)traversLayer:(LOTCompositionContainer *)layer withLayerName:(NSString *)layerName
{
    if ([layer.layerName isEqualToString:layerName]) {
        return layer;
    }
    else if ([layer isKindOfClass:[LOTCompositionContainer class]]) {
        for (LOTCompositionContainer *sublayer in layer.childLayers) {
            LOTLayerContainer *targetLayer = [self traversLayer:sublayer withLayerName:layerName];
            if (targetLayer) {
                return targetLayer;
            }
        }
    }
    return nil;
}
}
```

## 0x3 同层元素交互

iOS点击响应主要依赖Hit-Testing事件的传递链，因此只要在点击LottieView时，判断点击区域在自定义View的Layer内(LottieView视图层级均为Layer，不嵌入自定义View本身)，将Hit-Testing事件直接传给Layer的delegate自定义View，由自定义View继续在其内部传递Hit-Testing事件即可。

如上所述，问题的关键在于“点击LottieView时，如何判断点击区域是否在自定义View内”。

### 1.点击区域判断

- 方案1 使用iOS系统方法-(CALayer \*)[CALayer hitTest:CGPoint]

使用系统方法能确保准确找到触摸的图层，问题：当自定义View的Layer之上有透明图层或者带透明区域的png图片图层时，自定义View不能命中查询(能看到按钮但不能点击响应，违反用户直觉判断)。

```
- (UIView *)hitTest:(CGPoint)point withEvent:(UIEvent *)event
{
    CGPoint newPoint = [self convertPoint:point toView:self.superview];
    CALayer *layer = [self.layer.presentationLayer hitTest:newPoint].modelLayer;
    UIView *view = (UIView *)layer.delegate;
    if ([layer.delegate isKindOfClass:[UIView class]]) {
        return [view hitTest:newPoint withEvent:event];
    }
    return [super hitTest:point withEvent:event];
}
```

- 方案2 只遍历添加的自定义View，判断CGPoint包含关系

如代码所示，逆序判断添加的自定义view，即当触摸点在多个自定义view内时，后添加的自定义view接收该点击。问题：当自定义view被可见的lottie图层遮挡，但触摸点在自定义view内时，误判为自定义view需要响应该点击(看不到的按钮响应了该点击，违反用户直觉判断)。

```
- (UIView *)hitTest:(CGPoint)point withEvent:(UIEvent *)event
{
    NSArray *reversedItems = [[self.mixLayers reverseObjectEnumerator] allObjects];
    for (CALayer *layer in reversedItems) {
        CGPoint newPoint = [self.layer.presentationLayer convertPoint:point toLayer:layer.presentationLayer];
        if ([layer.presentationLayer containsPoint:newPoint]
            && [layer.delegate isKindOfClass:[UIView class]]) {
            return [(UIView *)layer.delegate hitTest:newPoint withEvent:event];
        }
    }
    return [super hitTest:point withEvent:event];
}
```

- 方案选择

经过内部讨论，我们采用了方案2。而“看不到的按钮响应了该点击”的问题解决，一方面由动效设计师保证可交互控件只当其处于最上层时才移入舞台，另一方面由研发同学在接收该点击的逻辑中兜底处理，判断当前

Lottie动画所在的播放帧(NSNumber \*)LOTAnimationView.currentFrame, 决定是否要执行响应逻辑。

## 2.透明控件处理

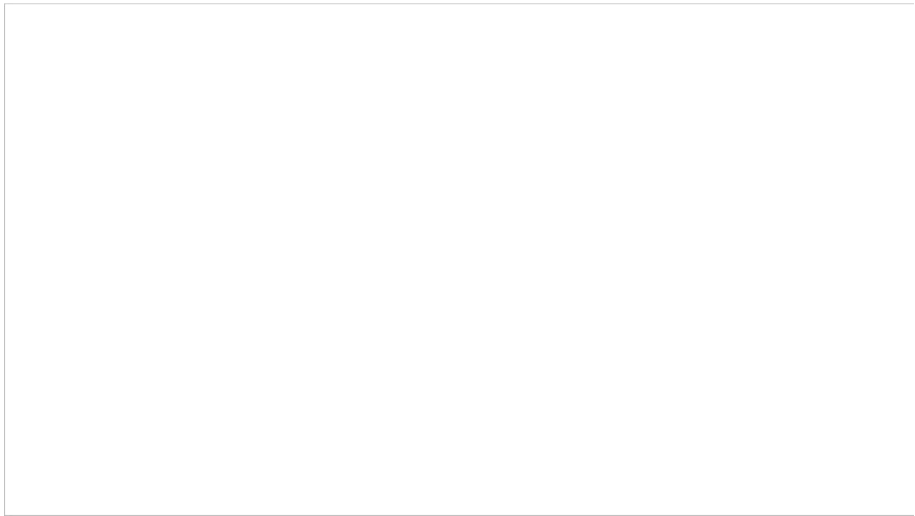
在实际使用中, 我们遇到了一个问题, 动效设计师在按钮点击完成后, 仅将按钮透明度设成了0, 但位置和大小都没有变化。因此如下代码中, 我们增加了透明度的判断。

ini 复制代码

```
- (UIView *)hitTest:(CGPoint)point withEvent:(UIEvent *)event
{
    NSArray *reversedItems = [[self.mixLayers reverseObjectEnumerator] allObjects];
    for (CALayer *layer in reversedItems) {
        CGPoint newPoint = [self.layer.presentationLayer convertPoint:point toLayer:layer.presentationLayer];
        CALayer *origLayer = layer.superlayer.superlayer;
        BOOL ignore40pacity = NO;
        while (origLayer != self.layer && origLayer.allowsGroupOpacity) {
            if (origLayer.opacity < 0.1) {
                ignore40pacity = YES;
                break;
            }
            origLayer = origLayer.superlayer;
        }
        if (!ignore40pacity
            && [layer.presentationLayer containsPoint:newPoint]
            && [layer.delegate isKindOfClass:[UIView class]]) {
            return [(UIView *)layer.delegate hitTest:newPoint withEvent:event];
        }
    }
    return [super hitTest:point withEvent:event];
}
```

## 0x4 总结

### 1.实现示例



以上视频中, 左侧为Lottie原始文件, 右侧为同层实现。可以看出, 即使逐帧比较, 自定义添加的View跟其它动画元素也有精密的轨迹同步, 并且其中的按钮控件能正常响应点击操作。

## 2.使用场景的限制

- 当触摸点在多个自定义view内时, 后添加的自定义view接收该点击。
- 当自定义view被可见的lottie图层遮挡, 但触摸点在自定义view内时, 可能误判为自定义view需要响应该点击(看不到的按钮响应了该点击, 违反用户直觉判断)。如有此种情况, 业务要在target-action中根据动画的当前播放帧, 判断是否要执行响应逻辑。
- 目前仅验证了UIButton按钮点击, 在自定义view中注册target-action即可, 代码编写与原生开发相同。

## 3.后续计划

- 探索是否有更好的点击区域处理的方案
- 验证更丰富的交互场景, 例如长列表、播放器等

hi, 我是快手电商的格蓝

快手电商无线技术团队正在招贤纳士!!!! 我们是公司的核心业务线, 这里云集了各路高手, 也充满了

机会与挑战. 伴随着业务的高速发展, 团队也在快速扩张. 欢迎各位高手加入我们, 一起创造世界级的电商产品~

热招岗位: Android/iOS 高级开发, Android/iOS 专家, Java 架构师, 产品经理(电商背景), 测试开发... 大量 HC 等你来呦~

内部推荐请发简历至 >>>我们的邮箱: [hr.ec@kuaishou.com](mailto:hr.ec@kuaishou.com) <<<, 备注我的花名成功率更高哦~ ☺