# LoadingView组件功能介绍

## 功能介绍：

**前言：**

对于一些简单的动画我们开发者可以使用系统提供的一些方法来实现，但是在实际开发中设计师给出的动画都是很炫酷复杂的，如果采用手写代码的方式，就要面对很多问题：

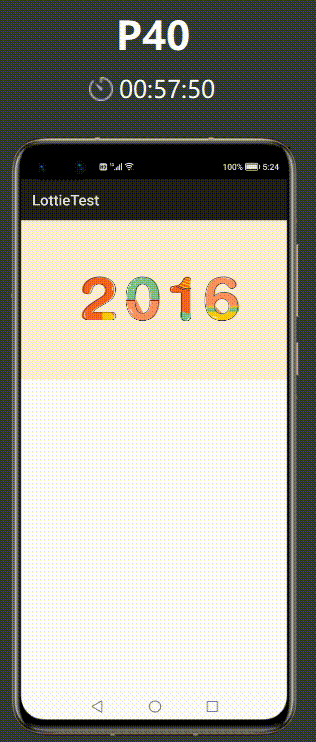
1. 如不同平台要重复开发，
2. 开发者和动画设计师之间的沟通问题，
3. 复杂动画对应的代码也非常复杂，后期维护困难等。

**Lottie功能介绍：**

Lottie是Airbnb 专门为移动开发设计的一个第三方开源库，他的优点如下：

1. 跨平台（目前支持Android、iOS 、Web、React Native等平台），本组件完成了Lottie再HarmonyOS上的移植。
2. 设计师通过After Effects将动画导出JSON文件，然后由Lottie加载和渲染这个文件并转成相应的代码，由于是JSON文件，文件也会很小，可以减少App包的大小。
3. 把动画制作和APP开发的工作分开，由设计师来完成动画的制作

## 模拟器上运行效果：





# **Lottie使用方法**

## 新建工程，增加组件Har包依赖

在应用模块中添加HAR，只需要将lottie.har复制到entry\libs目录下即可（由于build.gradle中已经依赖的libs目录下的\*.har，因此不需要在做修改）。

## **增加动画json文件**

在resources/rawfile目录下，放入动画对应的json文件，如bullseye.json.

## **修改主页面的布局文件**

修改主页面的布局文件ability\_main.xml，在Layout标签中增加app命名空间。

然后添加一个com.airbnb.lottie.LottieView组件，设置LottieView\_jsonFile属性。

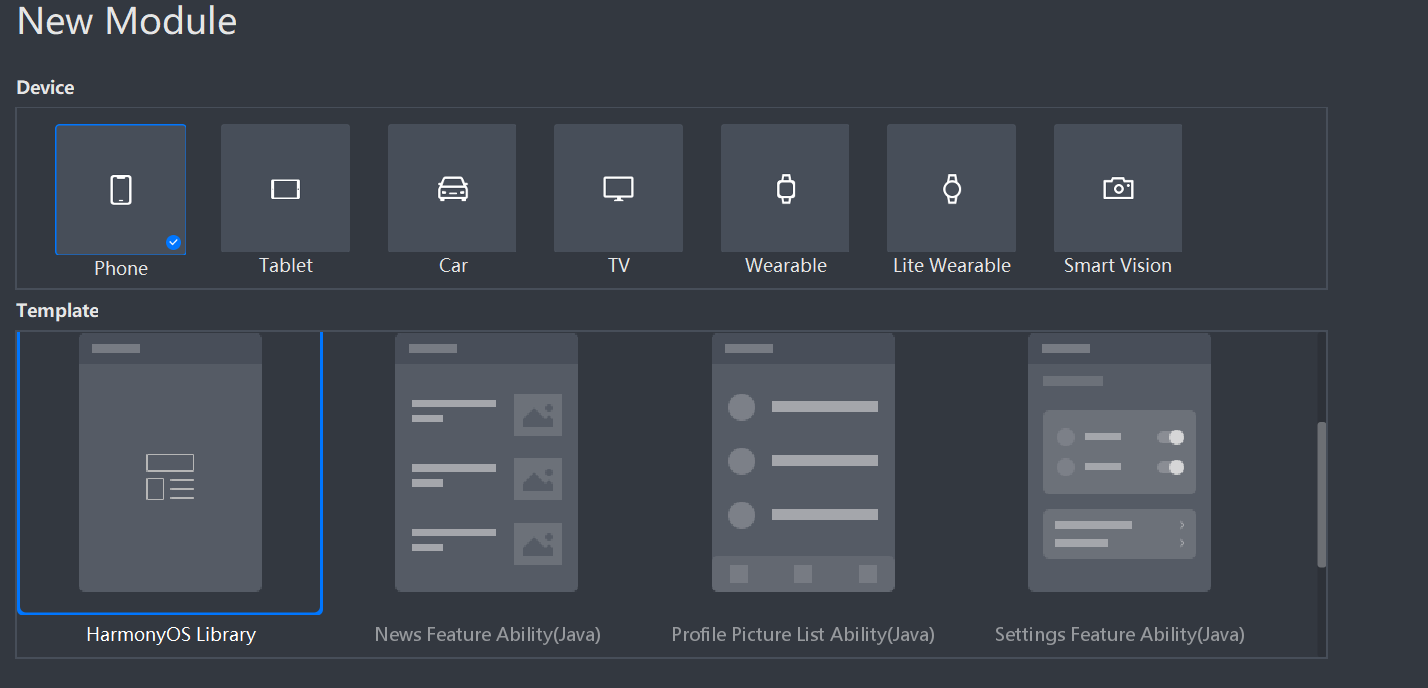
全部代码如下:

<?xml version**="1.0"** encoding**="utf-8"**?>  
<DirectionalLayout  
 xmlns:ohos**="http://schemas.huawei.com/res/ohos"** xmlns:app**="http://schemas.huawei.com/apk/res/ohos"** ohos:height**="match\_parent"** ohos:width**="match\_parent"** ohos:orientation**="vertical"**>  
  
 <com.airbnb.lottie.LottieView  
 ohos:id**="$+id:lottieView"** ohos:height**="match\_content"** ohos:width**="match\_content"** app:LottieView\_jsonFile **= "resources/rawfile/bullseye.json"** />  
</DirectionalLayout>

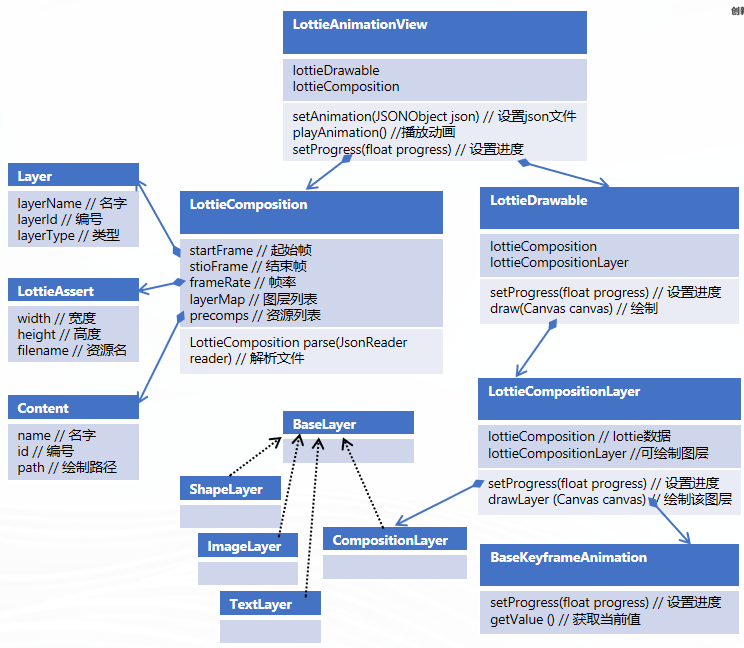
# **Lottie开发实现**

## **新建一个Module**

新建一个Module，类型选择HarmonyOS Library，模块名为lottie，如图



## **主要类介绍**



## **JSON解析流程介绍**

// 解析json文件，获取InputStream  
ResourceManager resourceManager = getContext().getResourceManager();  
RawFileEntry rawFileEntry = resourceManager.getRawFileEntry(jsonFile);

resource = rawFileEntry.openRawFile();

// 解析json文件,返回lottieComposition  
JsonReader reader = JsonReader.*of*(*buffer*(*source*(resource)));

// 构造compositionLayer  
compositionLayer = new CompositionLayer(  
 null,  
 LayerParser.*parse*(lottieComposition),  
 lottieComposition.getLayers(),  
 lottieComposition);

其中json文件大致说明如下：

{

"w", // 0 宽度  
"h", // 1 高度  
"ip", // 2 其实帧  
"op", // 3 结束帧  
"fr", // 4 帧率  
"v", // 5 版本号  
"layers", // 6 图层列表  
"assets", // 7 图片资源列表  
"fonts", // 8 字体列表  
"chars", // 9 字体具体样式  
"markers" // 10 遮盖层列表

}

其中assets图片资源说明如下：

"assets": [

{

"id", // 0 图片id  
"w", // 2 宽度  
"h", // 3 高度  
"p", // 4 图片名称  
"u" // 5 图片路径

}

]

layers说明如下：

"layers": [

{

"nm", // 0 名称  
"ind", // 1 索引  
"refId", // 2 指向的资源id  
"ty", // 3 图层类型(0:复合型，1: Solid类型 2:图片类型 4:Shape类型 5:Text类型)

parent", // 4 父图层  
"sw", // 5 Solid宽度  
"sh", // 6 Solid高度  
"sc", // 7 Solid颜色  
"ks", // 8 包含的动画  
"tt", // 9 遮盖类型  
"masksProperties", // 10 遮盖列表  
"shapes", // 11 图层包含的形状元素  
"t", // 12 文本属性  
"ef", // 13 填充效果(Lottie doesn't support layer effects)  
"sr", // 14 时间调整系数  
"st", // 15 图层起始帧  
"w", // 16 图层宽度  
"h", // 17 图层高度  
"ip", // 18 图层起始关键帧  
"op", // 19 图层结束关键帧

}

]

layers中shapes解析如下：

"shapes": [

{

"ty", //类型(如gr:图形组, fl:图形填充, sh:图形路径)

"nm",//名称  
"hd", // 是否隐藏  
"it" // 图形的轨迹集合

[{  
"ind",//索引  
"ks", //动画关键帧

{

"t", // startFrame  
"s", // startValue  
"e", // endValue  
"o", // 动画起始点坐标  
"i", // 动画结束点坐标   
"h", // 插值器类型  
"to", // 7 路径切线1坐标  
"ti" // 8 路径切线2坐标

}

}]

}

]

## **播放流程介绍**

Lottie源码播放调用栈梳理如下：

LottieDrawable.draw()

drawInternal()

drawWithOriginalAspectRatio()

baseLayer.draw()

compositionLayer.drawLayer()

baseLayer.draw()

compositionLayer.drawLayer()

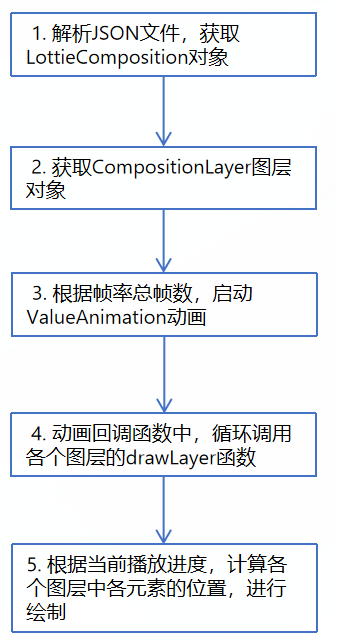
baseLayer.draw()

ShapeLayer.drawLayer()

ContentGroup.draw()

FillContent.draw() or StrokeContent.draw()

## **主流程介绍**



## **编译HAR包**

利用Gradle可以将HarmonyOS Library库模块构建为HAR包，构建HAR包的方法如下：

在Gradle构建任务中，双击PackageDebugHar或PackageReleaseHar任务，构建Debug类型或Release类型的HAR。

待构建任务完成后，可以在工程目录中的loadingview> bulid > outputs > har目录中，获取生成的HAR包。

# **项目源码**

见github代码仓：https://github.com/isoftstone-dev/Lottie\_HarmonyOS