# 鸿蒙开源第三方组件

——哔哩哔哩开源弹幕库的使用

jingle bells~

jingle bells~

jingle all the way~

圣诞节到了，我来给大家送圣诞礼物啦~

那就是鸿蒙的开源三方组件 - 弹幕库



本周开始，我们将兑现承诺，逐步开源一些我们的成果，首先给大家带来的是鸿蒙开源第三方组件系列01——弹幕库的使用和sample讲解。

今天向大家介绍的弹幕库组件是基于github上哔哩哔哩开源的项目（<https://github.com/bilibili/DanmakuFlameMaster>）所移植的。当前我们已经在鸿蒙系统上实现了弹幕的隐藏、显示、暂停、继续、发送、定时发送弹幕等一系列的功能，并开源了我们移植的源码（https://gitee.com/iscas-ohos/danmaku-flame-master\_ohos）。

为了更加清晰的介绍弹幕库的作用和使用方法，本文将从基础知识、使用方法、 sample解析三个方面进行讲解。

Round 1 弹幕库的基础知识

弹幕库是一个开源的三方库，在正式讲解弹幕库的使用之前，我们先来了解下第三方库的含义。第三方库是指除本地类库、系统类库以外的类库，需要后来安装，才能调取使用。

简单来讲，相当于其他人写好了代码，实现了一些功能接口，我们直接调用这些接口，就可以实现一些我们想要的功能。

举个例子：在一个应用中实现视频缓存功能（断网后可继续播放提前缓存的内容），如果想要自己实现的话，需要调用鸿蒙/安卓底层的方法，代码量非常大。如果已经有第三方库可以做到这点，并且开源了出来，那么你只需要引用他的第三方库，然后调用此库内公开的一些方法接口，简简单单的几行代码就可以实现缓存功能。

了解了什么是第三方库，现在我们来看弹幕库组件。

看视频时，弹幕是必不可少的互动功能，这个功能就是调用了弹幕库组件来实现的。现在我们已经实现了弹幕库功能在鸿蒙平台的构建，接下来我们一起深入了解下鸿蒙平台的弹幕库吧！

Round 2 弹幕库的使用方法

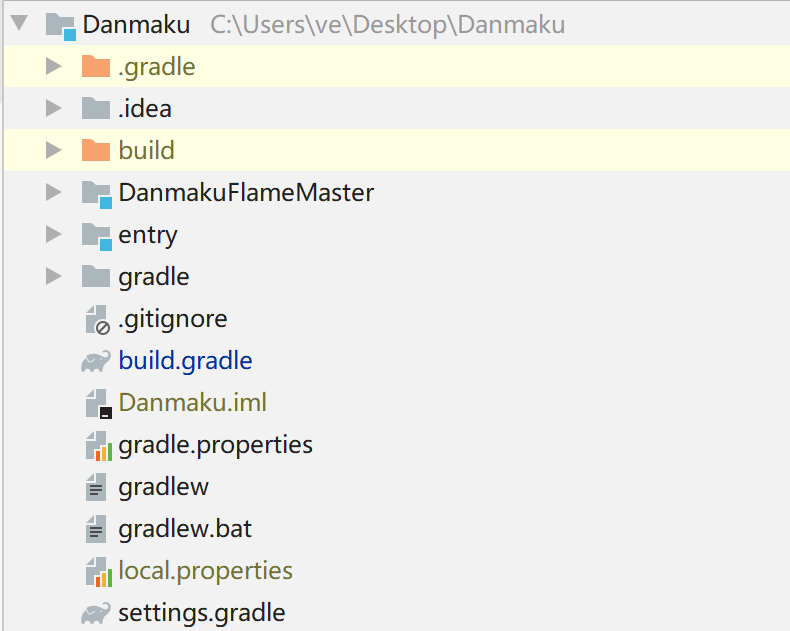
关于弹幕库第三方组件的使用方法，我们准备了视频教学版和文字教学版，大家根据流量情况使用哦~

视频教学版链接：https://www.bilibili.com/video/BV1Dy4y1i7cG/

（<弹幕库的使用文字版.docx>）

Round 3 sample解析

导入弹幕库组件后，其整个项目的文件结构如下：



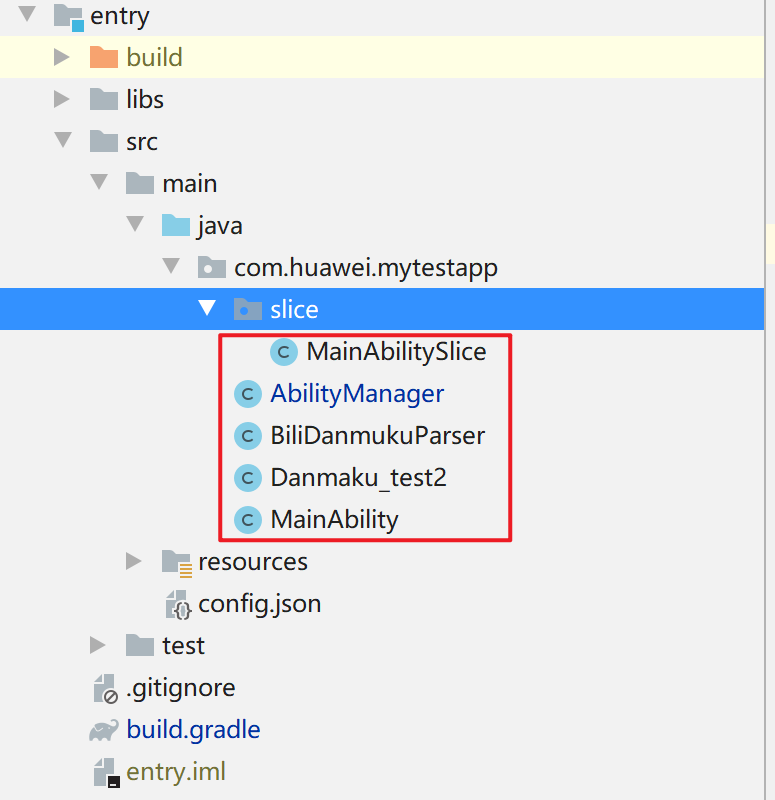
整个项目有2个module：entry和DanmakuFlameMaster。

在这里大家可以简单的理解：

* “DanmakuFlameMaster”就是之前提到的【第三方库】，它里面实现了弹幕功能的 所有细节。
* entry就是一个具体的“客户端”module，它声明了在鸿蒙程序中的所有运行细节。
* 最后跑起来的程序就是entry，entry里面实现了所有功能，包括界面、按钮、功能等等，只不过entry调用了第三方库“DanmakuFlameMaster”，使用了它的能力来实现了一些弹幕相关的功能。

本篇文章主要介绍如何在已经构建好了“DanmakuFlameMaster”这个第三方库的基础上，来进行“entry”的开发。

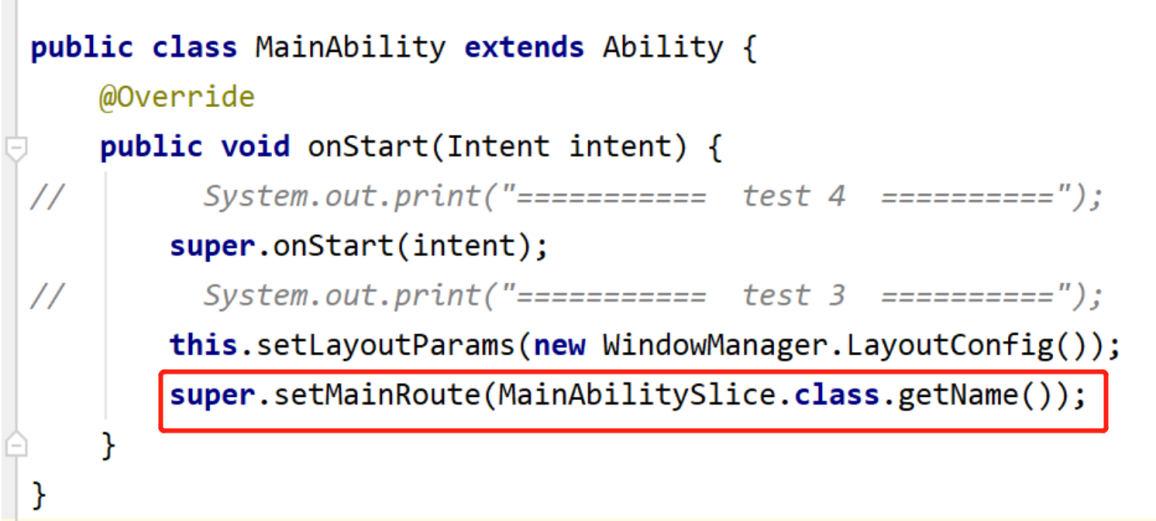
为了节省篇幅，此处我们使用一个已经开发完成了的sample，接下来以这个sample为例，向大家讲解弹幕库组件的应用及效果。



将entry展开之后，我们可以看到，真正的代码是真的很少，主要的功能都集中在MainAbility和MainAbilitySlice文件中，加起来的代码量也就不到1000行。

接下面我们对这两个文件进行逐个解析。

一、 MainAbility，它是一切应用的入口。



根据鸿蒙能力，这里是可以通过setMainRoute()方法调用很多的Slice来进行功能的实现，上图标记的代码就是声明了MainRoute在MainAbilitySlice之中。

二、MainAbilitySlice，这个文件比较重要。

这个文件主要包括：**import文件、初始化函数、功能实现函数**三个部分。

1、import 文件

此处import的文件可以分为三类：SDK文件、应用内文件和三方件文件。

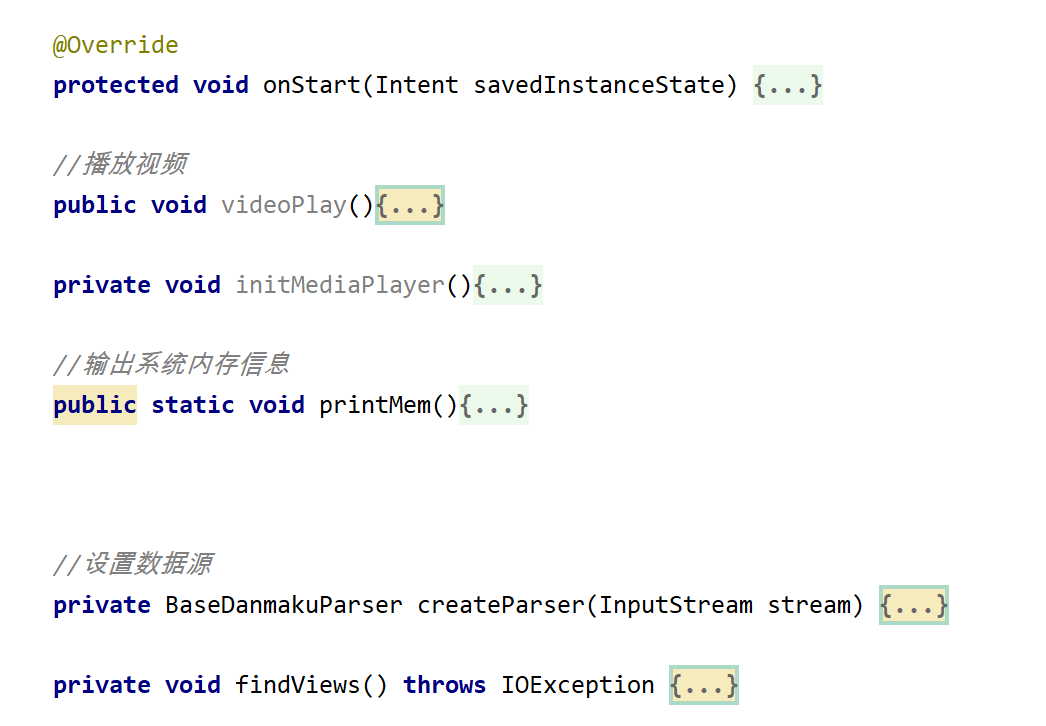
SDK文件： import后路径为ohos.....的。

应用内文件：import后路径为com.huawei......的。

三方库文件：import后路径为master.flame.danmuku......的。



2、初始化函数：



如图所示，初始化函数一共包含六个，我们将其分为五个部分，这五个部分与图中函数的顺序对应如下：UI初始化、视频源的设置与播放（2个函数）、输出系统内存信息、设置数据源（这里的数据指的是弹幕数据）以及弹幕输出。这五个部分之中，我们主要讲解其中的与弹幕库密切相关的部分：UI初始化、弹幕数据源与弹幕输出。

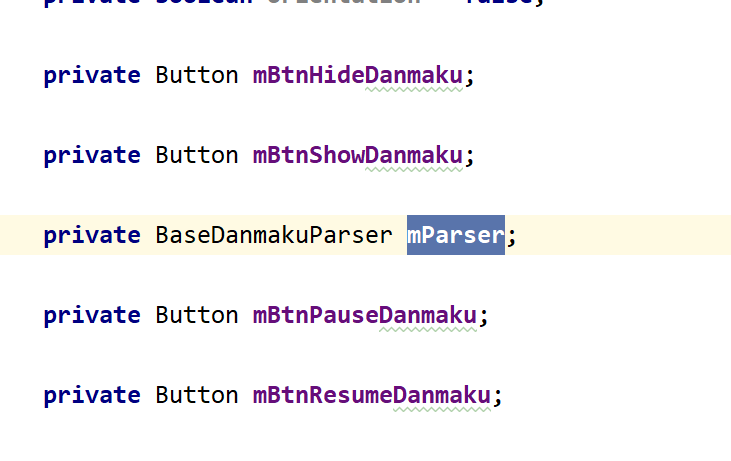
1）onStart方法实现了UI初始化



以上代码构建了弹幕库的显示窗口 设置了屏幕参数的读取 。

1. 弹幕数据源和最终的弹幕输出

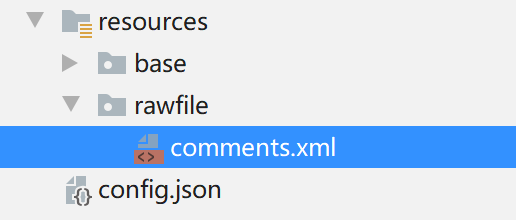
这两个部分有强相关，我们放在一起来介绍。



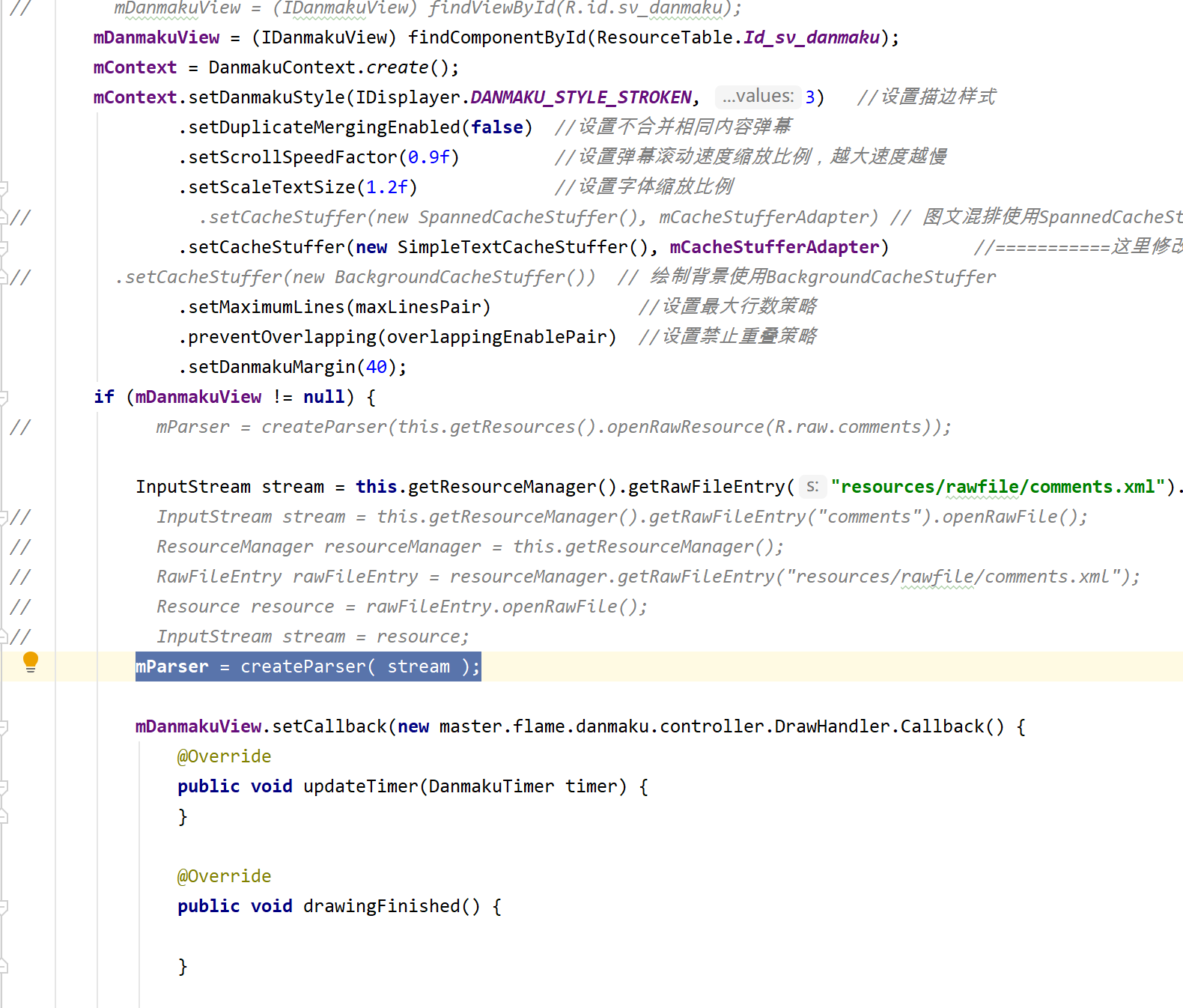
第一步：声明全局变量 mParser用来承载弹幕数据。



第二步：在加载数据源处调用了我们的第三方库的BiliDanmukuParser类，然后从我们的resources中读取弹幕数据：



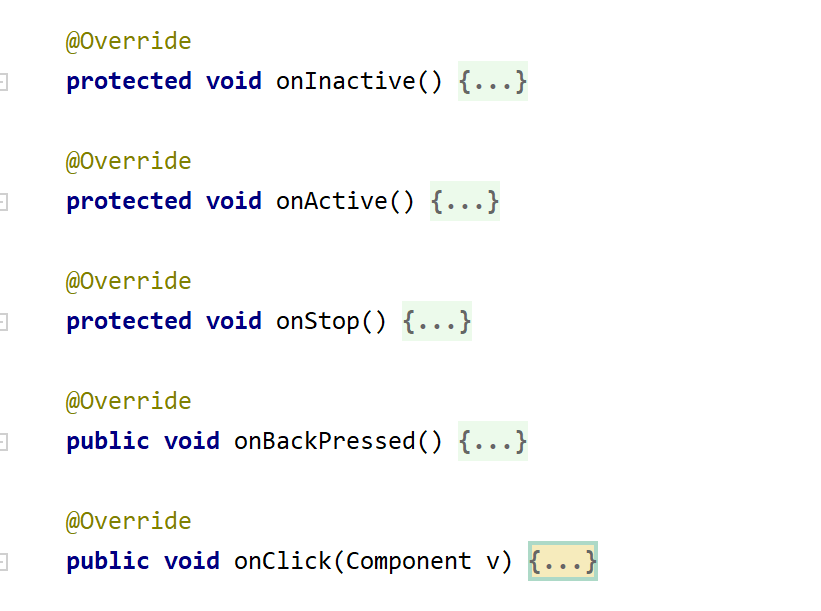
可以看到在datasource其实就是读取了这里的文件，大家可以自己点进去看一下这个文件内容，里面就是我们的“弹幕数据库”。



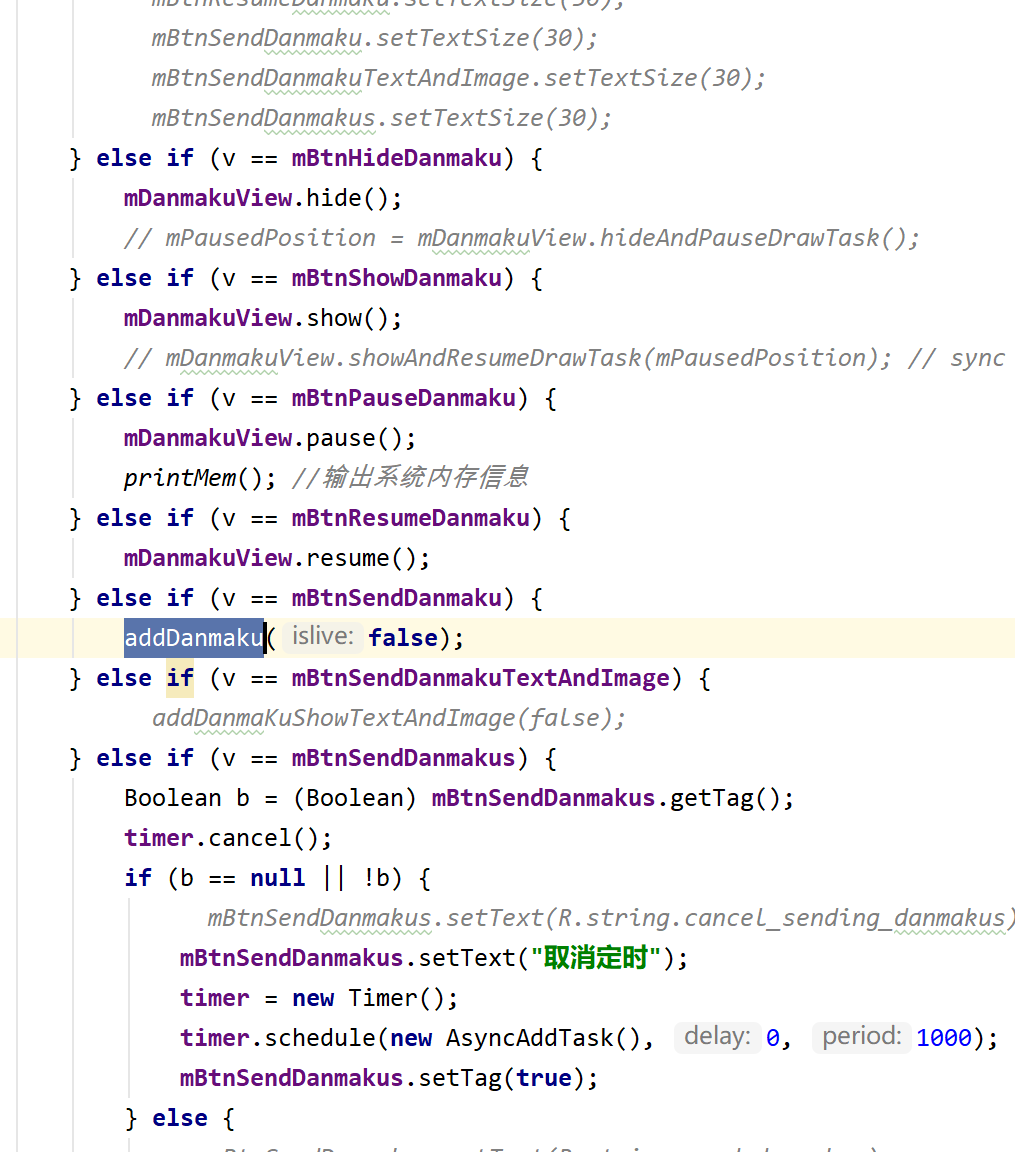
第三步：将载入了弹幕的BaseDanmakuParser类对象，return给全局变量mParser

（如果大家想找到这部分代码的话，可以到findview()中去寻找）。

3、功能实现函数



在UI上可见的一些按钮，每个按钮都对应一个callback函数，点击按钮，执行对应的函数体，实现相关功能。



上图所示为点击不通的按钮，出现不同的弹幕效果，例如：隐藏弹幕、显示弹幕、暂停弹幕等。

如果大家对三方库“DanmakuFlameMaster”更感兴趣，可以继续关注我们后续的内容，在之后的内容中我们会详细的教大家如何实现一个三方库的移植，分享“DanmakuFlameMaster”在实际移植过程中遇到的一些困难和解决方案 。



点个关注不迷路，内容更新更速度~