痛点:

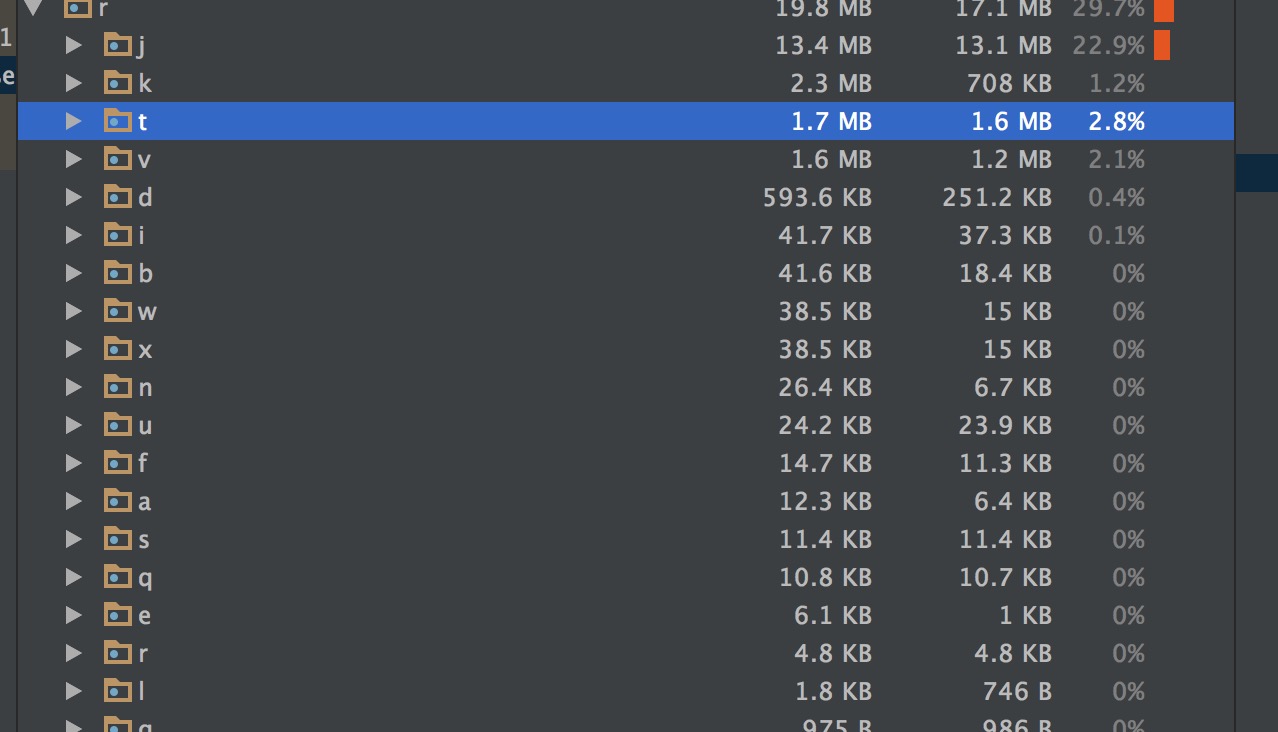
每次发版apk包太大，非常耗流量

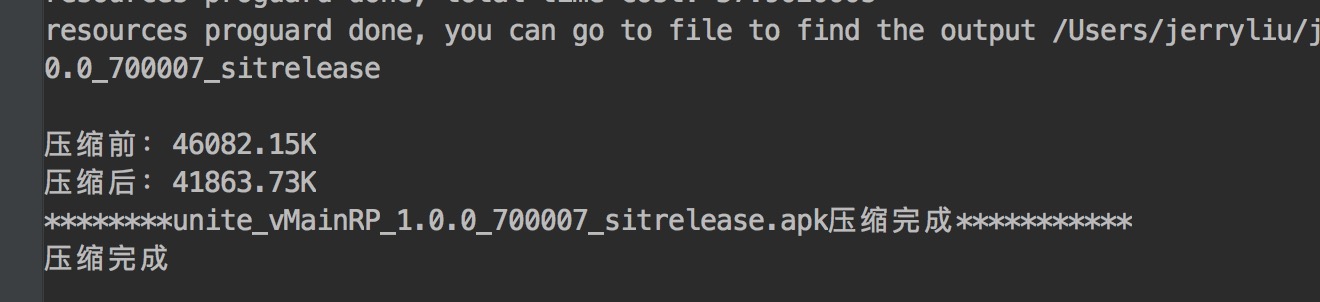
解决方案:

一 .针对整体优化

1. 对apk进行压缩： 采用AndResGuard压缩apk资源

效果如下：res文件夹已经变成了r，里面的各个资源也全部更换成其他的别名





mainrp压缩前46082k ~46.26M. 压缩后:41863k ~40.88M 可减少5.38M

二.针对资源优化

1：通过lint移除无用的资源

2: Png图片压缩:可以通过使用图片压缩工具对png图片进行压缩,例如tinypng

3大背景图处理：纯色图+后台下载的方式很好的解决了这个问题，客户端先使用纯色图片，然后大图从后端下载，这样只是启动的前几次使用纯色图，以后都会使用大图。

4:support libray语言资源：通过gradle配置只保留中文和英文

5: So优化:只保留"x86", "armeabi-v7a”, 考虑so做成插件

三．其它资源策略：

1.首先考虑能否不用图片，比如使用shape代码实现。

2.其次如果用图片的话，能否优先使用.9图来简化图片。

3.如果图片只是旋转角度或者颜色不同，可以用代码实现变换。

痛点:

同城代码和收派很多业务代码耦合在一起，存在大量的if else判断，耦合比较严重

解决方案:

重构同城和收派耦合部分的代码，采用接口的方式各自处理各自的业务逻辑，解除耦合