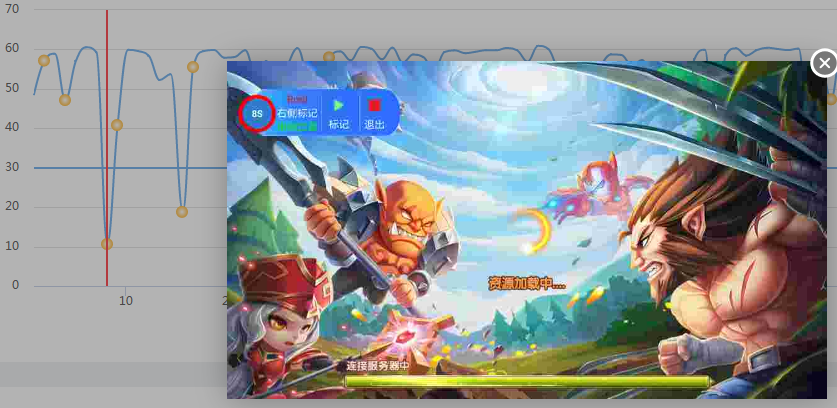
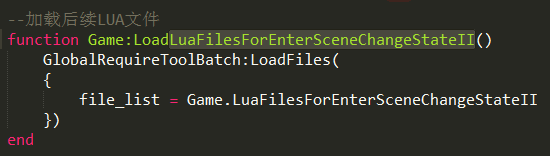
卡顿问题



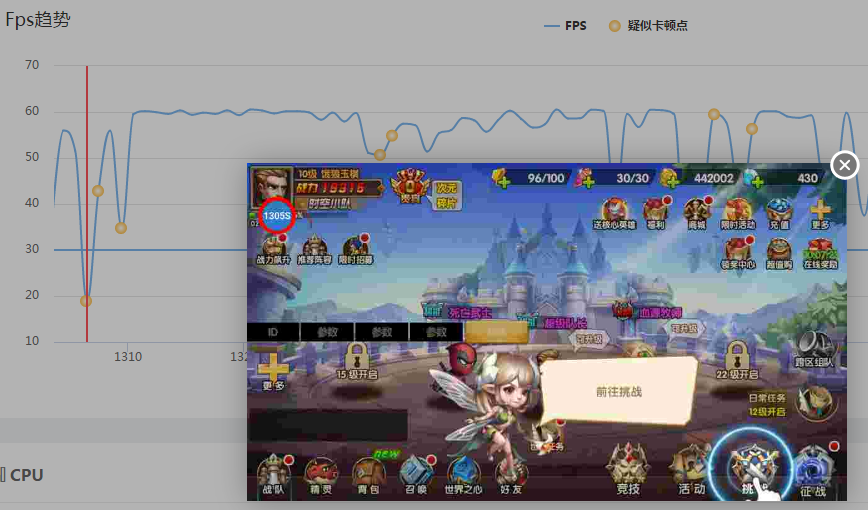
定位如下：



解决方案：

1. 拆分此处加载的LUA大文件。
2. 优化RequireToolBatch降低FPS波动。

卡顿问题



定位如下：

大部分FPS瞬时下降，都出现在打开界面、切换场景、播放大型特效，由于drawcall和三角形数维持在高水平的时候，FPS也只是维持在相对低水平，并不会出现瞬时下降。

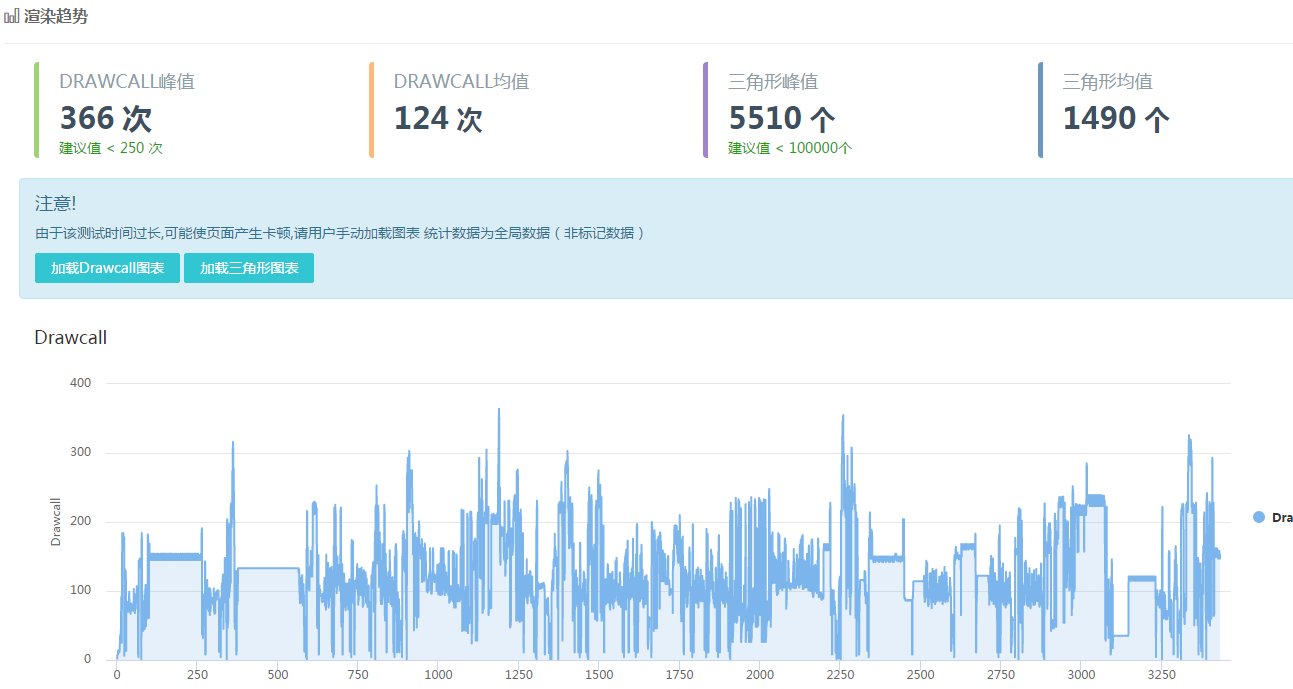
目前主要怀疑几个点：

1. 在主线程调用glTexImage2D提交纹理耗时较长
2. 在主线程调用CreateTemplateByData解析并创建粒子特效模板耗时较长
3. 副本地图尺寸过大且没分块，提高了加载耗时

解决方案：

1. 使用压缩纹理，在主线程调用glCompressedTexImage2D提交纹理
2. 在子线程调用CreateTemplateByData解析并创建粒子特效模板耗时较长
3. 副本地图分块加载

Drawcall峰值过高



定位如下：

主要是由于UI界面的层级结构过于复杂，无法发挥引擎的batching机制。

解决方案：

参考UGUI的batching机制，在cocos2d-x的Widget类中，实现针对UI的batching机制。

https://www.cnblogs.com/zhaoqingqing/archive/2018/09/25/9658403.html