# **XY地图容量分析和优化**

作者：韩宇飞

链接：<http://gad.qq.com/article/detail/45244>

**本文首发于知乎专栏：[MACK的游戏开发笔记](https://zhuanlan.zhihu.com/p/38260354/edit)，欢迎各位关注。**

也是一篇很久之前，在11年写的一篇分析和优化笔记，当时的目标也是为了大幅减小客户端包体的容量，这只是很多轮优化中的一次，最终我们还是做到了在100多张地图的情况下整个客户端安装包的容量在1.6G以内。  
  
**Gmcf优化**  
从Gmcf统计工具可以看出，草和Lightmap以及AOmap在整个Gmcf中占据了大部分空间。具体占据多大的百分比，各个地图各有不同。例如带两套uv的物件多的场景中光照贴图就多一些。草刷的多的地图，gmcf中草的容量就大一些。一些列出了几个地图的gmcf数据。  
  
**1） 新手村**  
文件头大小：0.00392151M  
全局数据块大小：6.03552M  
所有Chunk包围盒大小：0.0111237M  
所有地表数据大小（去除LightMap和AOMap后）：10.5365M  
所有模型数据大小（去除LightMap和AOMap后）：1.72444M  
**所有草的大小：11.127M**  
所有水的大小：0.228012M  
**所有LightMap和AOMap大小：23.7335M  
总大小（包含LightMap和AOMap）：53.4M**  
  
**2） 轩辕城**  
文件头大小：0.00392151M  
全局数据块大小：6.03572M  
所有Chunk包围盒大小：0.0140762M  
所有地表数据大小（去除LightMap和AOMap后）：13.4672M  
所有模型数据大小（去除LightMap和AOMap后）：2.76089M  
**所有草的大小：28.9539M**  
所有水的大小：0.687012M  
**所有LightMap和AOMap大小：55.0773M  
总大小（包含LightMap和AOMap）：107.0M**  
  
**3） 黑水城**  
文件头大小：0.000991821M  
全局数据块大小：1.51791M  
所有Chunk包围盒大小：0.00585938M  
所有地表数据大小（去除LightMap和AOMap后）：5.22597M  
所有模型数据大小（去除LightMap和AOMap后）：0.648087M  
所有草的大小：3.42282M  
所有水的大小：0.537189M  
**所有LightMap和AOMap大小：8.8412M  
总大小（包含LightMap和AOMap）：20.2M**  
  
**4） underground\_NLQ1**  
文件头大小：0.000991821M  
全局数据块大小：1.52285M  
所有Chunk包围盒大小：0.00263214M  
所有地表数据大小（去除LightMap和AOMap后）：2.6362M  
所有模型数据大小（去除LightMap和AOMap后）：1.06586M  
所有草的大小：0M  
所有水的大小：0.00043869M  
**所有LightMap和AOMap大小：27.27103M  
总大小（包含LightMap和AOMap）：32.5M**  
  
**5） 炎火城**  
文件头大小：0.000991821M  
全局数据块大小：1.53514M  
所有Chunk包围盒大小：0.0058136M  
所有地表数据大小（去除LightMap和AOMap后）：5.50868M  
所有模型数据大小（去除LightMap和AOMap后）：0.965645M  
所有草的大小：1.98259M  
所有水的大小：1.56363M  
**所有LightMap和AOMap大小：20.1375M  
总大小（包含LightMap和AOMap）：31.7M**  
  
从以上数据可以看出，gmcf中的草和光照贴图是占的大部分。  
  
**正在实施的方案有两个：**  
**一、 对于光照贴图：**

在导出gmcf的时候把所有地表和物件的光照贴图分别导出成若干张dds文件，然后用第三方工具对这些dxt1的dds文件单独压缩。可以省下一定的空间。  
  
**二、 对于草的数据：**

由于当前地图编辑器导出的草的数据是把草的所有几何数据导出（顶点数组、UV数组、索引数组、贴图名）。所以容量比较大。目前的解决方案是在地图编辑器中修改导出草的数据。

**每棵草用4个浮点数表示。第一个浮点数存Rotate，第二个浮点数存当前草在chunk中的XY索引，即x / 100.0f + y。第三个浮点数存Z，第四个存缩放。**  
例如一个chunk有三种不同类型的草，a草M棵，b草N棵，c草K棵。则该chunk草的导出的总大小为：  
(M+N+K)\*4\*4字节 == (M+N+K)\*16  
  
而原先导出的每种草中的：  
顶点数组大小：N\*4\*3\*4字节（每个草四个顶点，每个顶点大小为sizeof(NiPoint3)）  
UV数组大小：N\*4\*2\*2\*4(每个草4个顶点，每个顶点2套uv，每个uv为一个sizeof(NiPoint2))  
索引数组大小：N\*2\*3\*2(每个草两个三角形，每个三角形三个索引，每个索引为一个sizeof(unsigned short))  
  
**所以原先导出一个chunk草大小为：(M+N+K)\*124  
初步算下来如果按新的方式导出，草的总大小可以减到原先的1/7.75左右。**  
**但是所有的草的几何数据需要在客户端加载chunk的时候计算。会消耗掉一些性能。**