

游戏《被尘封的故事》钻石修改分析

不会知道是突发奇想还是游戏无聊，去找到了一款游戏-《被尘封的故事》，游戏最早来自 MC 夏日风情模组。游戏嘛最重要的是欢乐 早期一点点玩过，但是这个游戏可玩性实在太差，不想肝，只能破解氪金了

游戏修改呢有很多办法嘛，最好用的就是修改器了，由于游戏是安卓端的所以首选 [GG 修改器](#)。

GG 修改器呢十分简单，一般拿来简单修改内存中的数据就可以了，但是，这个游戏钻石（需要氪金的游戏货币）居然不在内存中存储，内存中只有一个可以显示的数据。所以一直在写入存档？抑或直接读写存档？

游戏存档数据很多啊，但是直接读写效率肯定下降啊。所以肯定采用的其他方法，这里不分析游戏数据存档，只分析安卓传统数据存储

安卓数据存储方式一共五种分别是

1. 使用 SharedPreferences 存储数据
2. 文件存储数据
3. SQLite 数据库存储数据
4. 使用 ContentProvider 存储数据
5. 网络存储数据

游戏采用的是第一种存储方案

游戏包名为：com.mindblocks.blocks_light

在/data/data 目录下可以找到 com.mindblocks.blocks_light 文件夹

在该文件夹下看到有好几个文件夹其中 shared_prefs 文件夹用于存储一些数据 打开这个文件夹

文件夹下有两个.xml 文件

```
com.heyzap.sdk.ads.xml
com.mindblocks.blocks_light.v2.playerprefs.xml
```

从名字分析

第一个应该是 SDK 接口一类;

第二个是某种数据;

查看第二个文件

搜索数值（当时是 90），

然后看到一行代码：

```
<int name="com.outlinegames.unibill.currencies.diamond.balance" value="90" />
```

看名字好长 但是有一个单词 “diamond” 钻石,并且后面的值就是“90”,不过为了安全起见我还是先去再次游戏不断刷怪得到了几个钻石🎉;然后再次查看,发现的确跟着更新了才确定;

在 java 中 int 类型最大为 “-2147483648” 到 “2147483647” ;所以修改也尽可能不要超出这个范围

进入游戏测试发现游戏数据达到了预期效果，修改成功

以上就是《被尘封的故事》中钻石修改的心得了 同样还可用于其他将金币等数值使用 SharedPreferences 存储数据的情况