逆向gfx tool 第二天

笔记本:	gfx逆向		
创建时间:	2019/10/10 10:25	**更新时间:**	2019/11/9 14:57
作者:	381643188@qq.com		
URL:	https://www.cnblogs.com/liweis/p/4653496.html		

背景

海外版的吃鸡老下载不下来，所以没法验证game booster功能是否有效。手机上安装了王者，使用gfx tool对王者分辨率进行设置，分别为960（图一）和2560（图二），结果没有差异。

尝试新apk

为了节约时间，直接下载了gfx tool，这个可以对国内外版本的吃鸡都有效果。包名为eu.tsoml.graphicssettings，对这个apk进行逆向。

先定位apk的路径。

```
adb shell pm path eu.tsoml.graphicssettings
package:/data/app/eu.tsoml.graphicssettings-F00gXe2vi_6tR3cNv8TAvA==/base.apk
```

进行dump定位主界面为eu.tsoml.graphicssettings.MainActivity。界面比之前简洁很多。

使用工具验证有效性。图一是默认分辨率，图二是修改分辨率为2560 * 1440，经过对比，**肉眼可见已经生效。**

逆向

使用jadx工具直接反编译，然后导出gradle工程。
搜索文件**应用设定**，id为accept。定位出按钮在布局fragment_gfx.xml中，按钮类型为MaterialButton，id为accept.
点击事件在eu.tsoml.graphicssettings.b中，变量名M0。点击事件在同文件中定义，内部类为class o implements OnClickListener。

点击类中代码较多，所以先跟踪分辨率的逻辑。搜索2560 x 1440，资源名为resolution的资源数组。

```
<array name="resolution">
    <item>默认</item>
    <item>768x432</item>
```

```
    <item>960x540</item>
    <item>1024x576</item>
    <item>1280(HD)</item>
    <item>1366x768</item>
    <item>1440(HD+)</item>
    <item>1600x900</item>
    <item>1920x1080</item>
    <item>2560x1440</item>
</array>
```

发现会出现在两个布局中，这时需要通过其他元素精细化定位，这里选择了字符串**设置游戏最佳分辨率**，定位到布局在items.xml中。
选择器类型为spinnerGraphics，id为spinnerGraphics。发现定义在eu.tsoml.graphicssettings.b中，变量名为Y。发现有两个地方会对spinner进行读取操作。

- public void B0() { 这里有些文件操作，但是具体目录地址在反编译过程中损坏，无法直接看出。
- public void T() { 这里记录到SharedPreferences里边了，应该只是记录用户的操作记录。

所以对public void B0()进行重点突破
这里通过jeb对eu.tsoml.graphicssettings.b进行反编译，找到方法B0()，这里他对分辨率选项进行了转义，转义过程如下。

- 默认 v0=a.a(77) v1_1=this.h1 然后走a(v0, v1_1)
- 768x432
- 960x540
- 1024x576
- 1280(HD)
- 1366x768
- 1440(HD+)
- 1600x900
- 1920x1080
- 2560x1440 v0=a.a(77) v1=86 v1_1=a.a(v1) a(v0, v1_1)

我们跟单条路径追踪，2560x1440，这里需要对a.a进行精细化处理，先跳过。看一下需要修改的文件。

a(v0, v1_1)中对sdk卡文件进行了读写。文件路径为

```
private String K0() {
    return Environment.getExternalStorageDirectory().getPath() + this.d1;
}
```

d1根据游戏类型被赋值为同步的地址：

```
this.d1 = a.a(273);
this.d1 = a.a(277);
this.d1 = a.a(281);
this.d1 = a.a(285);
this.d1 = a.a(289);
this.d1 = a.a(293);
this.d1 = a.a(297);
```

最终还是要对a.a进行精细化处理，可以获得最后的路径。这个时候工具就不太灵了，有三个字符串数组反编译无法获取到正确结果，这里就要通过阅读smali代码进行分析。关键语法。

```
aput-object vx, vy, vz
将vx的对象引用作为元素存入对象引用数组，数组的引用位于寄存器vy，元素的索引位于寄存器vz。

new-array vx, vy
vx为创建的结果，vy为数组元素个数
```

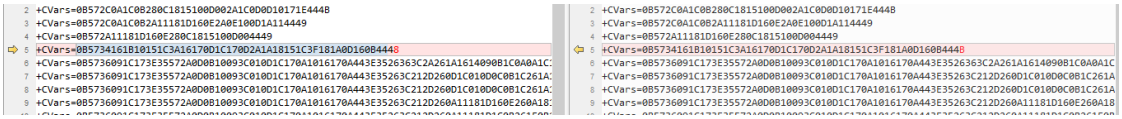
最终代码见github，反编译结果路径为，都在sd卡上：

```
 jiamiaohe: item = 77, result =
0B5734161B10151C3A16170D1C170D2A1A18151C3F181A0D160B44
jiamiaohe: item = 86, result = 4B
jiamiaohe: item = 273, result =
/Android/data/com.tencent.ig/files/UE4Game/ShadowTrackerExtra/ShadowTrackerExtra/

 jiamiaohe: item = 277, result =
/Android/data/com.tencent.tmgp.pubgmhd/files/UE4Game/ShadowTrackerExtra/ShadowTra

jiamiaohe: item = 281, result =
/Android/data/com.pubg.krmobile/files/UE4Game/ShadowTrackerExtra/ShadowTrackerExt

jiamiaohe: item = 285, result =
/Android/data/com.vng.pubgmobile/files/UE4Game/ShadowTrackerExtra/ShadowTrackerEx

jiamiaohe: item = 289, result =
/Android/data/com.rekoo.pubgm/files/UE4Game/ShadowTrackerExtra/ShadowTrackerExtra

jiamiaohe: item = 293, result =
/Android/data/com.tencent.iglite/files/UE4Game/ShadowTrackerExtra/ShadowTrackerEx

jiamiaohe: item = 297, result =
```

```
/Android/data/com.tencent.igce/files/UE4Game/ShadowTrackerExtra/ShadowTrackerExtr
```

同时写入的内容和a.a(86)、a.a(77)有关，翻译后

```
v0=0B5734161B10151C3A16170D1C170D2A1A18151C3F181A0D160B44
v1=86
v1_1=a.a(v1)=4B
a(v0, v1_1)
```



左侧是默认，右侧是分辨率2560x1440。这就是gfx tool能修改游戏参数的原因。

分辨率的显示内容为：

- 默认
- 768x432
- 960x540
- 1024x576
- 1280(HD)
- 1366x768
- 1440(HD+)
- 1600x900
- 1920x1080
- 2560x1440

对于分辨率
前半部分是77。
后半部分如列表。

列表显示	实际值	加密值
默认	81	48
768x432	78	49574F49
960x540	79	49574E4C
1024x576	80	49574149
1280(HD)	81	48
1366x768	82	4857494F
1440(HD+)	83	4857484B
1600x900	84	48574A

列表显示	实际值	加密值
1920x1080	85	48574C
2560x1440	86	4B