

手机性能评测 --3D 场景

阅读 424 收藏 11 2016-08-26

原文链接：www.jianshu.com

想知道如何成为一名机器学习工程师，并将预测模型应用于各大领域内的大数据处理中吗？Udacity 优达学城纳米学位将有最适合你的课程 <http://cn.udacity.com/mlnd>

概述

书接上回[手机性能评测--2D场景](#)，这次讲讲3D评测。随着AR/VR方兴未艾，手机势必会成为各个厂商和开发者的一块战略要地。目前暂露头角的要属Pokémon GO（一款抓宠物小精灵的AR游戏）了。那怎么知道我们的手机是否能流畅的运行3D程序呢？其实和2D测评大同小异，都是获取每一帧的绘制时间。

疑问

根据2D场景的测评可以知道，使用gfxinfo可以抓到graphic信息，那3D场景继续使用该方法不就可以了么？当我们满怀信心的在3D场景下执行adb shell dump gfxinfo后，却无法得到任何信息。why?!!!!因为准确的来说，gfxinfo所对应的service只能获得**2D view**的绘制时间，要获取3D场景的信息需要使用其他的手段。



android手机上，不论2D还是3D的绘制都需要SurfaceFlinger的参与，既然glxinfo无法获取3D的绘制，那我们就直捣黄龙去拿surfaceflinger里面的信息。根据 TesterHome社区中的[统计fps](#)文章，可以知道，循环使用

```
adb shell dumpsys SurfaceFlinger --latency
```

命令，然后处理每次命令的结果就可以得到fps

注：这里是拿的fps。 即是将每帧的绘制时间转换成了fps。

命令的运行结果如下：

```
# The first line is the refresh period
# A) when the app started to draw
# B) the vsync immediately preceding SF submitting the frame to the h/w
# C) timestamp immediately after SF submitted that frame to the h/w
16954612
7657467895508 7657482691352 7657493499756
7657484466553 7657499645964 7657511077881
7657500793457 7657516600576 7657527404785
(...)
```

第一行表示：刷新时间间隔，第二行第一列表示：app里数据（bitmap/material等）准备时间第二行第二列表示：cpu将数据传递给h/w（hardware）绘制时的同步时间。（这里可以理解为cpu和gpu绘制一帧时，二个线程的同步时间）第二行第三列表示：h/w（hardware）绘制完成的时间后面的行和第二行一样，会dump出128帧的数据，也



这些信息和gfxinfo中dump出的**Draw Process Execute**十分相似。虽然有人写了python脚本和shell脚本，对以上命令进行了包装，可以方便的获取fps，但是想作为测评软件来说，是不可行的。因为我们想要的结果是点击运行按钮后就可以显示出每帧的绘制时间，而不是运行脚本。

解决方案

既然我们已经知道了获取3D场景每帧绘制时间的方法，那我们就可以写一个app来包装这个方法，达到简易测评的目的。由dumpsys实现原理可以知道，

```
adb shell dumpsys SurfaceFlinger --latency
```

命令的实现，其实就是调用了surfaceFlinger服务的dump方法而已。那我们就先来找一下surfaceFlinger服务的dump是在哪里实现的

dump SurfaceFlinger实现包装

文件：`/frameworks/native/services/surfaceflinger/main_surfaceflinger.cpp`中有如下代码：

```
int main(int, char**) {  
    ...  
    // instantiate surfaceflinger  
    sp flinger = new SurfaceFlinger();  
    ....  
    // publish surface flinger
```



```
}
```

文件：/frameworks/native/services/surfaceflinger/SurfaceFlinger.h

```
static char const* getServiceName() ANDROID_API {  
    return "SurfaceFlinger";  
}
```

可以得知的是main_surfaceflinger的main函数运行（也就是SurfaceFlinger服务启动前的入口）的时候，调用了ServiceManager的**addService**方法，将SurfaceFlinger.cpp注册为了名字为"SurfaceFlinger"的服务。所以最后的实现其实是SurfaceFlinger.cpp的dump方法。等等？！！SurfaceFlinger.cpp？？.cpp？？根本没有办法像**封装gfxInfo一样**，使用java的反射机制来实现SurfaceFlinger的dump方法啊，并且surfaceflinger服务和我们的app属于不同进程。所以试图包装SurfaceFlinger的dump方法**走不通**。

shell脚本

网上有很多app demo可以执行shell脚本，那我们可以写一个类似的demo来执行

```
dumpsys SurfaceFlinger --latency
```

命令吗？不妨先试一试。

1. 先写一个shell脚本，hly.sh

2. push到data/local/tmp/目录

```
adb push xxx/xxx/hly.sh data/local/tmp/
```

确保获得了执行权限

```
ls -l data/local/tmp/hly.sh
```

3. 写一个app来执行hly.sh脚本。核心代码如下

```
process = Runtime.getRuntime().exec("sh");  
dos = new DataOutputStream(process.getOutputStream());  
for (String command : commands) {  
    if (command == null) {  
        continue;  
    }  
    dos.write(command.getBytes());  
    dos.writeBytes("\n");  
    dos.flush();  
}  
dos.writeBytes("exit\n");  
dos.flush();
```

其中commands的值是： /data/local/tmp/hly.sh

当你满心欢喜的期待正确结果的时候，却得到如下输出



wtf?!!!!!! , 只好再回过头来看看SurfaceFlinger.cpp中的dump方法

```
status_t SurfaceFlinger::dump(int fd, const Vector& args)
{
    ...
    if ((uid != AID_SHELL) &&
        !PermissionCache::checkPermission(sDump, pid, uid)) {
        result.appendFormat("Permission Denial: "
                           "can't dump SurfaceFlinger from pid=%d, uid=%d\n", pid, uid);
    } else {
        ...
    }
    ...
}
```

原来只能有**SHELL**或者配置得有**android.Manifest.permission.DUMP**权限的app才可以dump surfaceflinger的信息。又因为DUMP权限的配置必须是系统级别app，所以这条路也**走不通**

柳暗花明

至此，能想到的办法都走不通了。正当陷入三方app到底能否获取3D场景每帧绘制时间的时候，发现了一款十分厉害的三方软件**Gamebench**，不需要root，能获取4.1以上所有android系统的各类硬件，软件，性能信息。其中当然包括3D场景的fps。它是如何做到的？！！由于篇幅原因，请点击[Gamebench 3D场景FPS获取猜想](#)。

加入掘金
Android 交流微信群,
随时随地阅读干货, 交流见解。



相关热门文章

结束近两年的（安卓）工作有感

张拭心 13 2

通过修改 **LayoutInflater** , 全局替换字体!!!

承香墨影 34 1

项目需求讨论-仿ios底部弹框实现及分析

青蛙要fly 7 2

NullPointerException 与 Java 自动装箱拆箱机制

张拭心 3 1

Android 8.0 功能和 API

RXlee 32

