屏幕截屏并保存方案实现

DionysosLai 2014-5-27

**屏幕截屏思路大纲：**

1. 通过opengl来实现屏幕截屏，这种方法可以**完全复制你所看到的画面上的任何东西**。

关键函数是：glReadPixels(0, 0, w, h, GL\_RGBA, GL\_UNSIGNED\_BYTE, buffer);

1. 通过纹理对象来获取，可以获取当前对象的所有纹理信息。注意：**如果某些对象是不属于本对象，即使可以在屏幕上看到，但也无法获取下来**。

关键函数是：

//渲染纹理开始捕捉

renderTexture->begin();

//绘制当前场景

pCurScene->visit();

//结束

renderTexture->end();

二者获取图片如下所示：



Opengl 方法 纹理对象方法

注意左下角的不同。

1. 二者的优劣对比:
2. 图片大小: 通过opengl方法，获取的图片大小**小于**通过纹理对象来获取，大概有80k差左右（我只测过过《球类分分看》这个游戏。），而实际上前者的元素更多。
3. 效率对比：两种效率对比，前者的效率略高于后者（不过可以忽略）
4. 灵活性比较：opengl方法，只能很死板的复制像素信息；但通过纹理对象，可以做到先将屏幕缩小，然后在获取下来（我们在玩游戏时，游戏截屏通常都是整个屏幕的内容，但显示时，明显比较小-----就是通过这个方法来实现），同时也可以获取不必要的元素，比如游戏中的广告信息等。
5. 二者的应用领域：
6. 如果想看到的是游戏画面的所有元素，那么建议使用opengl，ios的屏幕快照，应该采用的就是这个方法（当然，ios在屏幕快照时，会将小圆圈先消失掉。）。如果你不想看到哪个元素，可以先让其消失，在截屏，最后又还原。
7. 如果是想看到游戏画面，诸如广告信息都不想看到，采用纹理方法，灵活性更强。
8. 使用屏幕截屏时注意点：

在使用屏幕截屏时，游戏的帧率会瞬间降低到很小，如果对于游戏帧率要求比较严格的游戏，比如游戏中，如果加入了box2d，那么在截屏瞬间会有世界失衡的感觉。

**图片保存思路大纲：**

使用引擎自带的saveToFile。主要问题在于不同平台保存路径相应有所差别，这个通过平台适配来来实习。具体代码如下：

#if (CC\_TARGET\_PLATFORM == CC\_PLATFORM\_ANDROID)

string filePath = "/sdcard/DCIM/Camera/" + path;

CCLOG("wanna save file path = %s",filePath.c\_str());

image->saveToFile(filePath.c\_str());

image->release();

return true;

#endif

#if (CC\_TARGET\_PLATFORM == CC\_PLATFORM\_WIN32)

string filePath = CCFileUtils::sharedFileUtils()->getWritablePath() + path;

CCLOG("wanna save file path = %s",filePath.c\_str());

image->saveToFile(filePath.c\_str());

image->release();

return true;

#endif

附录：

1. 在Android中加入文件读写权限：

在AndroidManifest.xml 加入如下代码：

<uses-permission android:name=*"android.permission.WRITE\_EXTERNAL\_STORAGE"*/>

<uses-permission android:name=*"android.permission.MOUNT\_UNMOUNT\_FILESYSTEMS"*/>

1. glReadPixels 函数解析：

函数原型：void glReadPixels(GLint x,GLint y,GLsizesi width,GLsizei height, GLenum format,GLenum type,GLvoid \*pixel);

函数参数(x, y)定义图像区域左下角点的坐标，width和height分别是图像的高度和宽度，\*pixel是一个[指针](http://baike.baidu.com/view/159417.htm)，指向存储图像数据的[数组](http://baike.baidu.com/view/209670.htm)。参数format指出所读象素[数据元素](http://baike.baidu.com/view/38785.htm)的格式（索引值或R、G、B、A值，如下面表所示），而参数type指出每个元素的[数据类型](http://baike.baidu.com/view/675645.htm)。具体参数如下所示：

+++++++++++[像素格式](http://baike.baidu.com/view/1220149.htm)表++++++++++++++++++++++

GL\_INDEX 单个颜色索引

GL\_RGB 先是红色分量，再是绿色分量，然后是蓝色分量

GL\_RED 单个红色分量

GL\_GREEN 单个绿色分量

GL\_BLUE 单个蓝色分量

GL\_ALPHA 单个Alpha值

GL\_LUMINANCE\_ALPHA 先是亮度分量，然后是Alpha值

GL\_STENCIL\_INDEX 单个的模板索引

GL\_DEPTH\_COMPONENT 单个深度分量

++++++++++++++++++++++++++++++++++++++++++

+++++++++++++++++++ [数据类型](http://baike.baidu.com/view/675645.htm)+++++++++++++++++++

GL\_UNSIGNED\_BYTE 无符号的8位整数

GL\_BYTE 8位整数

GL\_BITMAP 无符号的8位整数数组中的单个数位

GL\_UNSIGNED\_SHORT 无符号的16位整数

GL\_SHORT 16位整数

GL\_UNSIGNED\_INT 无符号的32位整数

GL\_INT 32位整数

GL\_FLOAT 单精度浮点数

+++++++++++++++++++++++++++++++++++++++