工作笔记

（去完善上一篇工作笔记）

1：unity编辑器的平台切换是干什么用的，不同的平台有什么特殊性？切换平台对既有工程有什么影响？调研一下

2：远程对象调用是个什么概念，应用场景有哪些，优缺点和替代方案有吗？

3：熟悉tk2d，工程里顺带学习如何通过tk2d制作UI

4：C#中new一个长度为0的数组，有何作用？

5：图片的png和jpg格式在游戏开发里的应用和区别，各自有没有什么优点或局限性？

6：项目里有一些Prefab在很多个场景里都有一份，假设：我在某一个场景里对该prefab作了修改并且apply和保存，那最新的修改会被同步到prefab吗？假设2：我在一个场景里单独操作了并修改了该prefab，那引用该prefab的场景里也会被同步吗？假设3：我在某个窗口prefab里操作并修改了该prefab，那引用该prefab的场景里也会被同步吗？

7：Canvas在Unity原生UI系统里代表什么，做什么用的？Unity原生的UI系统和NGUI相比如何，优缺点如何？

8：针对不同平台，做一些标签，这样在代码中可以灵活的开启或关闭一些代码，比宏命令还好用。这是C#语言的特性吗？

9：有空去了解下图文混排，请教下孟卿，了解下思路和一些比较难解决的点

10：都说在切换场景时，资源会被释放，但这部分资源是一个场景开始时，Unity根据场景构成和引用关系自动加载的资源，相当于是该场景的静态依赖资源，这部分资源只有在来回切场景和重置当前场景时才会被释放，所以在制作场景时尽量减少资源依赖，在场景玩法持续进行中，动态的加载资源与释放资源，这样既有利于管理资源也减少了场景依赖的资源数量。回头验证一下场景初始资源和场景中动态加载的资源之间卸载的区别

11：Resource.UnLoadAsset仅对通过Resource.Load方式加载到内存的资源有效，场景初始的资源不会被清理掉，写代码验证

12：PaintingRenderLayer类用于调整3d特效的层级，查看原理

13：播放影片是怎么实现的？学习一下

14：学习项目里的编辑器Inspector界面扩展

15：ugui下，场景的保存和Prefab的Apply是完全不同的两个概念和操作，作用的结果也是不同的，把一个Prefab拖到场景里，然后对这个Prefab再做一些增删改等操作，如果点击该Prefab的Apply，则在该场景下的所有改动都会被同步到Prefab里去，游戏里其他用到该Prefab的地方也会受到影响，一个场景里如果只是作为展示用到这个Prefab，则在改完后不Apply，而是直接保存场景，这样，既能保存在该场景，又不会影响到其他场景。

16：学习unity自带的事件系统和SendMessage的用法，利弊

17：学习unity的动画系统，3d和2d，2d的动画控制器(animator)

18：深入了解下unity的场景和prefab之间的依赖关系，prefab和prefab之间的依赖关系，是如何保存维护并且同步的？会不会出现问题依赖断开的情况，坑多吗？

19：ugui的锚点怎么实现

20：unity工程里的文本类的数据，出现过几次这边命名已经修改了，但运行游戏却看不到效果，一方面是没有执行打包流程，也就是说程序读取的和我们编辑的是两个文件，我们要做的是把编辑的文件实时同步到程序读取的文件，建议dc里专门一个文件夹放置数据类表格，任何改动都执行一次整体的打包操作

21：lethe项目里的的场景内特效和界面上的3d特效，是完全由美术操作的，之前出现过几次问题，材质丢失，应该是需要给美术提供一个规范的工作流程。

22：程序与资源（包括数据）的使用方式如何组织，才能使得，程序，美术，策划的分工妥帖，并且互不牵连，而且最大程度降低出错的几率。

23：prefab上关联了图片或者图集，加载prefab会自动加载图集

24：普通方式打一个界面的ab包，是否会把该界面所以来的图集全部包含进去？

25：前些天遇到一个问题：手机上重新装上游戏时，并用一个已经全部解锁的账号同步数据，返回到长河界面会播放解锁支线曲包的动画，跳到其他界面后重新返回到长河界面还是会播放特效，并且这个问题只有在新装的游戏第一次运行才会出。最初定位问题是放在了unity第一次运行时是否有一些特殊化，如果是按照这条线往下查可能会涉及到很深层次的知识，可按道理来讲unity也不会出现这类问题，还怀疑过是否是数据没有及时写入，因为看游戏表现像是数据没有及时写入的。写日志跟踪后发现后来的播放特效并不是代码控制的，而是GameObject在OnEnable时自主播放的，思路回归到了问题本身，之前就不应该怀疑是unity运行是否是第一次的问题，而是第一次播放过特效，问题和是否播放过特效有关，特效播放完后虽然是消失了但其Active还是true，返回长河后再次打开场合仍会播放特效，总的来说这次问题出在了播放特效的处理上，没有在播放完后的状态上处理得当，导致了隐患，这种隐患在后来的某一个很巧合的步骤上爆发了出来。

26：在打Android包的时候，出现了一个错误，提示合并AndroidManifest.xml文件出错。Unity直接打出Android包时合并AndroidManifest.xml文件有一个规则，在插件目录下的所有AndroidManifest.xml文件都会被合并成一个最终的，比如aar里的，比如jar里的，而Plugins/Android/AndroidManifest.xml文件是主的，其他的都往这个上合并。最初因为不知道合并原则，就胡乱删了一些东西，而且以为aar里需要依赖很多配置，所以改了好几次但还是起冲突。后来得知，aar里依赖一些必需的权限即可，application节点为空都可以，之前一直出错是因为Plugins目录下有两个小米的aar，引起问题的就是这个最老的，一遍一遍的看到出错，就时没有引起重视，想当然的以为是合并出了问题，可本身又不太清楚合并的具体规则，无法判断是否真的出错，就陷入了僵局。最后仔细看了出错的日志，才发现到底是哪个目录下的的xml出了错。最后：删除了最老的那个aar，删除了temp目录下的aar解压目录，最后问题解决。

这个问题前后持续了好几个小时，最后总结为：太想当然，以为问题是出在了自以为的地方，因为没找到对的方法所以一直在试错，茫无目的的。仔细追查出错日志，查看具体信息，到底是哪个目录下的xml出的错，因此深入到了问题本质。