1. config文件管理和调试(Platformentity.cs) (重点)

详细内容请查看: unity配置文档说明.doc

1. 打资源,打包,主要在packager.cs(同时也有测试打包资源相关内容), 同时要掌握调试阿里云上的资源项目 (重点)

对应内容查看下文的 资源打包与热更新流程记录

1. 热更新, 主要在staticutils.cs utils.cs, (重点)

对应流程查看下文的 资源打包与热更新流程记录

1. xcode导出后的编译 主要在 XCodePostProcess.cs,

用于方便打包人员打包, 减少出错率. 无特殊内容, 主要看代码.

1. 静默热更新(android 大包更新)主要在DownloadHelper.cs,

无特殊内容, 一般情况没有问题, 已经稳定的运行了4个版本更新. 主要看代码

1. socket底层 主要在socketmanager.cs, (重点)

对应内容查看下文的 Socket重连流程记录

1. 游戏整体运行流程 Gamemanager.cs (重点)

对应内容查看下文的 资源打包与热更新流程记录

1. 熟悉各种打包, iOS打包, android打包, mac版打包, ip测试包, 还有打包记录文档在Editor/Doc下

对应内容查看下文的 打包流程记录

对应的包的bundleid 及打包记录查看项目中的 平台打包记录.xlsx

1. lua项目接入

详细内容查看: lua项目使用说明.doc

1. 微信后台管理(跟微信登录,支付相关)

网址: <https://open.weixin.qq.com>

账户请查看: Mac电脑/Users/admin/Data/账号密码管理.txt

1. android插件源码

位于Assets/\_\_HappyCity/Editor/Plugin/WXPay\_AndroidPlugin\_Source.zip

1. 各个账号信息

详细请查看: Mac电脑/Users/admin/Data/账号密码管理.txt

1. 交互的相关人员

内容位于: Mac电脑/Users/admin/Data

Alien: 打包

andy\_me: pc端主管

coco: 北京的策划

galen: 运维

gigi: 美术主管

jack: 市场

keng: 广东的策划

lisa: 渠道推广

Mark: 各个账号注册等

Mars: 游戏后端

Leo: web端主管

1. ios打包证件管理

位于Mac电脑/Users/admin/Data/证件相关

1. 阿里云账号相关

详细内容请查看: 账号密码管理.txt

项目中文档路径: HappyCity\_lua\HappyCity\Assets\\_\_HappyCity\Editor\Docs

## 资源打包与热更新流程记录

1. file.txt 文件说明(资源版本管理文件)

{

"": {

"StreamingAssets": "5701b70fb33bb7b58a31c00411fa1e3a|11817|2"

},

"baselobby": {

"atlas\_avatar.assetbundle": "b65cde45cca0d2744f6105eac0bb0d64|1928512|1",

"basegame.assetbundle": "55aad3bdf796404943a6da7f5d748325|8081455|1"

},

..

"InstantUpdate\_VersionTime": 1486257913856,

"InstantUpdate\_VersionCode": 2

}

以上为简易的file.txt内容

灰色部分为模块名, 也是模块路径, "" 模块代表最外层的, 也就是StreamingAssets文件所在的路径,相对其他模块也就是空.

红色的为md5.

绿色的为文件大小.

蓝色的为资源版本号, 5.0以后的版本才有资源版本号.

InstantUpdate\_VersionTime 为版本时间, 就是打该版本的时候的时间, 一般小于该版本时间的不会进行热更新(也就是更新的时候不能后退更新, 同时可以用该值来控制版本不进行更新, 不过5.0以后的版本可以直接控制 版本控制文件(VersionControl.txt), 位于每个大版本号阿里云路径)

InstantUpdate\_VersionCode 热更新资源的版本号, 用于控制点击热更新的时候不会下载到错误的资源版本, 以免发生游戏不能正常运行的问题. 5.0以后才有的内容

1. 打资源流程(Packager.cs)
2. 获取当前的平台
3. 打assetbundle资源
4. 打lua资源(正式环境下会编译)
5. 通过资源的路径和md5等信息生产配置文件(file.txt) 的json信息(内存中)
6. 通过更新ali云的正式版本的资源配置文件, 比对后更新当前配置文件的资源版本号
7. 添加版本时间InstantUpdate\_VersionTime和资源版本号InstantUpdate\_VersionCode
8. 生产配置文件(硬盘上) 和 随包的配置文件(同样的内容, 用于点击热更新的时候比较需要下载的内容)
9. 抽取(拷贝其中的一部分)一份StreamingAssets目录下的资源
10. 抽取一份当前版本的资源到相对应的资源版本目录下
11. 压缩StreamingAssets下的lua资源到一个assetbundle(这步是为了加压的时候加速)
12. 热更新流程(Utils.cs, StaticUtils.cs, GameManager.cs 对于热更新包体的运行流程是先更新, 然后进入到loading后会启动热更新检测, 有需要热更新的内容就会下载, 下载后会在下次启动是进行更新. 所以 更新和下载是2个过程. 对于点击热更新(点击下载1个单独的游戏)则是1个过程, 下载了就直接在需要更新的位置了)
13. GameManager启动, 调用Init
14. 解压游戏(lua, 大厅资源, 其他资源 依次进行)( 为了加速, 大厅资源加压后就会加载界面, 并且可以进入到大厅)
15. 大厅解压后进行热更新调用DoInstantUpdate (把下载好的资源覆盖掉旧版本的资源)
16. 热更新后启动lua (到这里为止更新完成)
17. 第一个界面Loading启动后会调用Utils.cs 的StartInstantDownload 启动热更新(这里会判断是否是测试人员, 如果是测试人员,又是测试环境, 就好使用测试环境的资源进行热更新)
18. 之后会调用StaticUtils.cs 的 DownloadCloudAssets, 这里会进行错误判断, 如果有错, 就会重新启动该方法进行下载.
19. 之后会调用UpdateGameModules, 这里会调用CompareConfig进行版本比较,完成后调用UpdateFiles进行文件下载(使用WWWConfigUpdater.cs)
20. 最后调用UpdateConfig 修改配置文件(到这里,下载完成, 点击热更新的下载也基本相同)

## Socket重连流程记录

1. 调用SocketManager的Connect进行连接, 3个SocketConnect 都会同时进行socket连接, 首先使用第一个为默认的选中的连接. 启动m\_CheckTimeOut\_MonoAction, 用于检测连接超时
2. SocketConnect的socket连接后会启动线程进行消息接收, ReceiveData
3. 获取消息时会判断toke, token 不可用时会首先发送ShakeHands(握手协议), 然后调用OnStateConnected: 标记SocketConnect状态为已连接,去掉连接超时检测, 如果是选中的SocketConnect, 就会发送登录协议
4. 如果一个SocketConnect 断开, 会调用 DoDisconnect, 如果没有大于最大重连次数, 那么调用Reconnect进行重连, 否则标记成 NotConnected. 并通知SocketManager重连消息(onSocketDisconnectAction, OnSocetDisconnectHandler)
5. 如果是选中的SocketConnect断开了, SocketManager 会调用CheckConnectionState进行重连判断
6. 如果当前的SocketConnect中还有正常连接的SocketConnect, 就使用该SocketConnect
7. 否则, 查找状态不是NotConnect的SocketConnect, 如果有, 等待2秒 重新检测. 如果没有, 那么切换ip后调用SocketManager的Connect 进行重连.

## 打包流程记录

基本流程:

1.修过游戏名称 （注意：app store 版修改名称的时候要把 iTunes connect 里的名字也改过来）

2.修过平台 bundleId ， version，build （android的是 version code）

（注意：如果是企业版或者是ios app store版，要把player setting 中的Strip Engine Code 的勾去掉）

3.ios 企业版需要添加（\_IsEnterprise）， app store 版必须把这个去掉(现在没有企业版了)

4.捕鱼游戏添加(\_IsFish), 非捕鱼版必须把这个去掉

5.修过版本号（打包界面（2中的build）和GameEntityAll.cs的VersionCode）和版本名称（2中的version 和 GameEntityAll.cs的VersionName）, 还有 GameEntityAll 中的 bundleId （2中的bundleId）

6.修过PlatformGameDefine中的平台(现在已经可以直接切换平台)

7.修过平台对应的图片(BaseLobby/\_PlatformSwitch/Image) (现在已经可以直接切换平台)

8.发布ios app store 版本的时候要把第三方充值关掉(willing 关) (现在可以在热更新阿里云路径上的socket\_config.xml 里面 配置, 格式: qbq.u.game597=5.1 )

9.新包发布后要修改好version\_All.txt(version\_IOS\_All.txt)文件，使旧版本能提示大版本更新（willin 改）(现在version\_All\_Android.txt 或者ios 或者 捕鱼的跟游戏对应, 代码在GameEntity.cs)

10.企业版和android版提交到云端后把下载地址发给willin

简易流程(相对上面的流程做了简化):

平台切换 打包简化:(目前只有 597, 597wangwei, 510k, 131 可以使用, 单包无法使用)

1. 在PlatformSwitch目录下(Assets同级目录) 双击对应的平台

2. 修改 GameEntityAll.cs 中的bundleid, versionname, versioncode

3. unity加载完成后(unity右下角圈圈转完后), unity菜单栏中 点击 Game/PlatformSwtich/597(这里选对应的平台)

(以上3步完成(并且unity加载完)后再 打资源)

4. 如果是ios平台 在 热更新地址下 修改第三方充值

5. 如果是 android版本提醒willin修改version\_All.txt文件, 用于大版本更新(现在可以在热更新阿里云路径上的socket\_config.xml 里面 配置, 格式: qbq.u.game597=5.1 )

6. 提交热更新资源到阿里云

## 参数控制说明

1. Packager.cs
2. DefaultGameModuleList //用于控制StreamingAssets下的随包模块, 如果是空就是整包,如果模块以冒号”:”开头, 就是去掉该模块
3. m\_FixedVersionControlCode //(默认0)手动设置的 版本号, 如果该值大于0, 那么 m\_UpdateVersionControlCode 就等于该值;//用于跳过阿里云上某些有问题的版本号
4. m\_FixedVersionControlCode\_ali //(默认-1)手动设置的 参考用的版本号(原本应该是阿里云上正在用的版本号, 但是有些情况下该值也需要手动设置), 如果该值大于-1, 就不用阿里云上的VersionControl的值, 而是使用该值
5. m\_IsServerUrl //(默认true)是否使用服务端地址更新 配置文件. false 则使用本地配置文件//作用与GameManager中的该值相同, 区别是作用时机不同, Packager的该值是用于打包时(非运行时)获取配置文件, 而GameManager 中的值是用于运行时热更新
6. GameManager.cs
7. m\_IsServerUrl //(默认false)是否使用服务端地址更新 assetbundle, false 则使用本地资源
8. \_VersionControl\_ali //(默认0)用于设置本地测试用的 VersionControl 的版本号 该值大于0 才有效, 否则使用 阿里云上的VersionControl的值