**Xgame版本更新及sdk接入文档**

* **启动流程**

Lua层代码在checkVersion目录下，该目录下的功能不能依赖于游戏其他的文件，以防止更新游戏代码时把该目录下的东西给更新坏了。

1>android或ios平台启动游戏

2>lua层首先调用接口获取平台信息，如（mac，partnerId，imei，qn，checkVersionUrl，packageVersion，mobile，loginServerUrl）。

3>设置资源文件搜索路径android和ios的主搜索路径都为Resource/， ，增量脚本的搜索路径为程序版本号（即壳子版本号）前2位，假如版本号为1.2.3 ，则路径为1.2/updatedir

4>lua层调用各平台的sdk登陆接口，callStartSdkLogin，壳子上收到这个调用命令后，开始进行登录操作。

5>壳子层登陆完成后，调用Lua层的bitchStartGame(token) 方法，进入游戏内部登陆流程，token为客户端与服务器商定的用于校验的字段。

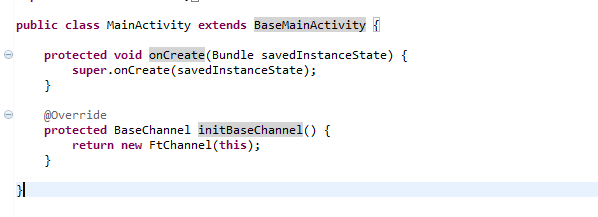
6>登陆完成后，lua层开始获取服务器列表并进入相应的游戏。

* **sdk接入 (android篇)**

1.在svn目录Client\android目录下创建工程，工程名请遵循如下规范:android--渠道简写，如梵丁的为 android-xgame-ft

2.在新建立的工程中包含ft\_xgame\frameworks\runtime-src\proj.android工程。

3.创建Channel类继承自BaseChannel。并实现需要实现的方法，方法说明请参照IChannel接口中的方法说明

4.创建mainactivity类，继承自BaseMainActivity。并初始化一个上一步创建的Channel类，如图所示: 

5.在AndroidManifest.xml文件中设置如下meta-data.



* **sdk接入(ios篇)**

该方案还没有开发完成，暂时只是在代码层实现了思路（大体思路跟android的差不多），只保障了程序编译没错，但由于机器的原因，一直没有真正运行起来过)

1.新建工程，工程放入

2.将frameworks\runtime-src\proj.ios\_mac工程拖入新建工程

3.实现协议GameMsgDelegate.h

4. LuaJni为lua层调用oc层或oc层调用lua层的api接口类

5. FtXgameBridgeInterface为GameMsgDelegate的代理类，初始化时要GameMsgDelegate设置进去。

* **遗留的一些问题**

1. 游戏中关于进入用户中心及切换账户及登出等接口都还没有实现，可参照VersionManager.startSdkLogin()中调用接口进行实现(不要修改该接口，可另行在外面定义一个接口来通用处理该消息，版本检测相关的文件不要与游戏逻辑产生关联)，现在android端已经定义好了接收消息接口，code定义如下:

//直接退出

**private** **final** **static** **int** *QUIT* = 1;

//吊起第三方的登陆界面

**private** **final** **static** **int** *SDK\_SHOW\_LOGIN* = 2;

//吊起第三方支付

**private** **final** **static** **int** *PAY* = 3;

//切换第三方账户

**private** **final** **static** **int** *ACCOUNT\_SWITH* = 4;

//进入sdk用户中心

**private** **final** **static** **int** *ENTER\_USER\_CENTER* = 5;

//注销sdk用户

**private** **final** **static** **int** *LOGIN\_OUT\_SDK\_USER* = 6;

ios接受消息的接口已定义好(LuaJni--- + (void) gameMsgDeal:(NSDictionary \*)dict)，但还未具体实现逻辑，code的定义也没写好，可与android定义成一样，然后以Notify的方式调用业务逻辑中的代码，你可以看到FtXgameBridgeInterface已经监听了相关的消息,如：

#define SWTICH\_ACCOUNT @"SwitchAccount"

#define OPEN\_USER\_CENTER @"openUserCenter"

#define GAME\_EXIT @"gameExit"

#define PAY @"pay"

#define LOGOUT\_PLAMT\_ACCOUNT @"logOutPlatAccount"

#define SDK\_LOGIN @"sdkLogin"

#define START\_GAME @"startGame"

1. Ios中LuaJni中的+ (NSDictionary \*) getPlatInfo方法中各个应该从配置文件中获取，版本号要获取程序版本号的前2位，这个版本号的获取跟增量更新有关，必须要传。
2. Ios的增量更新还没有测试过！！！！

* **增量更新及服务器列表选择功能详解**

此处说明为热血wp使用xgame增量更新及服务器列表相关功能而写。

-----》代码相关

1. lua层代码目录，在xgame的Resource下的checkversion目录下（注意为了保障更新木块的独立性，请不要把游戏代码写在该目录下，该模块不应该依赖游戏模块内容）
2. c++代码需要依赖Client\ft\_xgame\frameworks\cocos2d-x\extensions\assets-manager这个目录下的AssetsManager类及Client\ft\_xgame\frameworks\runtime-src\Classes下的lua\_assetsmanager\_test\_sample类。将assetsManager中的相关方法暴漏给Lua调用，可参照Client\ft\_xgame\frameworks\cocos2d-x\cocos\scripting\lua-bindings\auto中lua\_cocos2dx\_extension\_auto类中关于该assetsManager开放方法的相关代码。
3. 游戏启动的时候请先调用checkversion下的checkversion.lua文件。