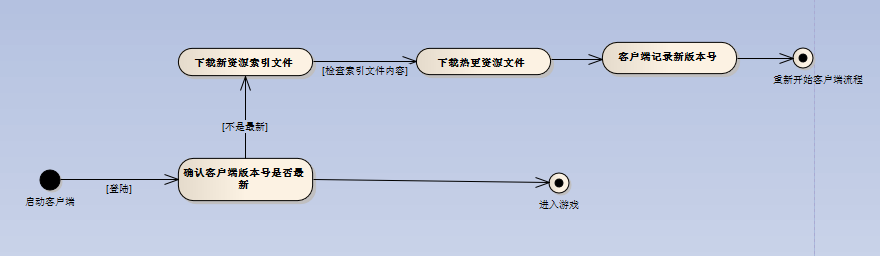
DGM客户端热更新方案

|  |  |  |  |
| --- | --- | --- | --- |
| **版本** | **更新时间** | **更新内容** | **更新人** |
| **V1.0** | 2016-03-02 | 建立文档 | 张青 |

# 热更新支持的内容

* 配置文件（config\shared\_config）
* UI逻辑代码（Hotfix.dll，可以支持多个UI界面的逻辑更新或者添加新界面）
* UI资源文件（UIPrefab,UIAtlas,Texture）
* 3d资源文件（动作\bp\挂件）->目前的框架中不包含，当前框架支持扩展。以后需要有。
* 音乐文件（歌曲mp3）

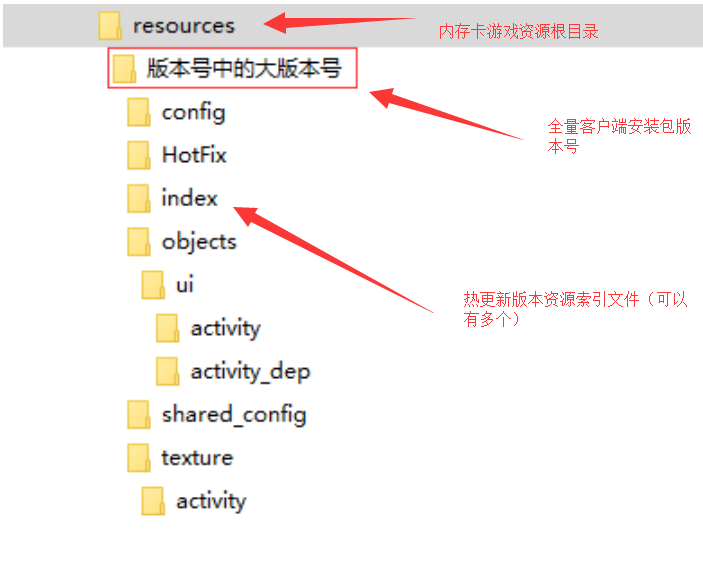
# 更新流程



# 版本定义

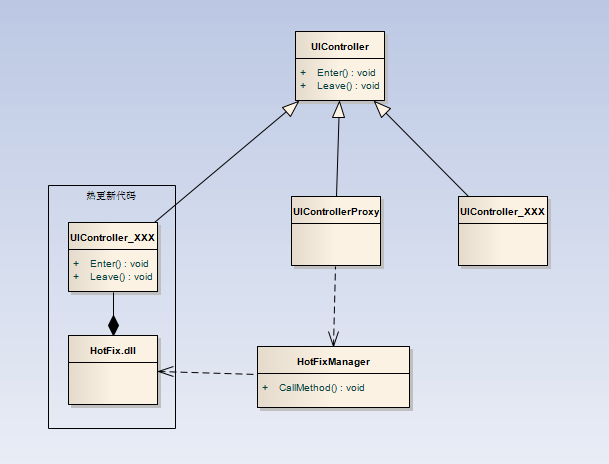
资源版本号 为字符串，表示为A.B.C，A为全量包版本，B为全量包下的资源版本，C为p4版本号。   
在游戏客户端有一个资源版本号，client\_res, 这个数据会有两个保存位置，第一全量包中会保存一份，第二当进行过资源更新后，在游戏客户端的本地文件中保存一份。   
cdn与服务端有一个资源版本号, server\_res。

# 客户端本地下载的热更新资源目录结构



* 以上目录为SD卡游戏目录下的用于热更新的目录结构。
* 当全量包安装后，会删除旧的大版本号目录，建立当前的大版本号目录
* Index目录用于存放资源索引文件，当存在多个热更小版本时，index目录下会保留当前大版本下的所有热更版本的资源索引文件；
* 存在多个热更小版本时，资源文件只保留最新版本——例如，1.1版包含A(1.1),B(1.1)资源。1.2版包含B(1.2),C(1.2)资源。则最终目录中包含A1(1.1),B2(1.2),C(1.2)资源及1.1和1.2的索引文件。
* 目前没有动作、bpt、挂件、音乐资源的热更新目录。可在当前框架下扩展

# 领域建模



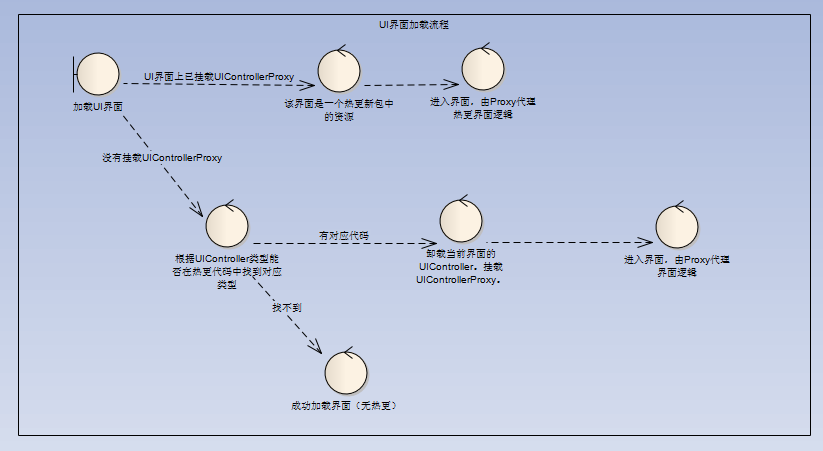
# 热更新代码加载流程

UI界面的逻辑代码写在UIController的派生类UIController\_XXX中；热更新涉及的界面的逻辑代码需要单独摘出到热更新工程中，输出成一个HotFix.dll文件,由客户端框架通过UIController的派生类UIControllerProxy代理执行操作。

载入一个UI界面时分为三种情况：

* 载入一个无需热更的界面
  + 界面资源在全量包中，挂在UIController\_XXX逻辑。
* 载入一个只热更新代码的界面
  + 界面资源在全量包中，同时HotFix.dll中包含该界面对应的代码UIController\_XXX。
* 载入一个热更新界面资源及代码的界面
  + 界面资源在热更新增量包中，界面上直接挂在UIControllerProxy，指向HotFix.dll中的代码
  + 界面代码在HotFix.dll中，通过UIControllerProxy代理调用。

加载流程如下：



# 其他

* **热更代码的执行效率不高，不适合对特别复杂的界面进行热更。**
* **热更新界面的交互事件使用代码添加，不是直接在面板上指定。**