子游戏开发需要遵循的规则:

ILRuntime版本为最新版.目前是2.0.2

Addressable版本为1.18.19 版本保持一致以免出问题.

假设ThisGameName 是本游戏确定的名字.

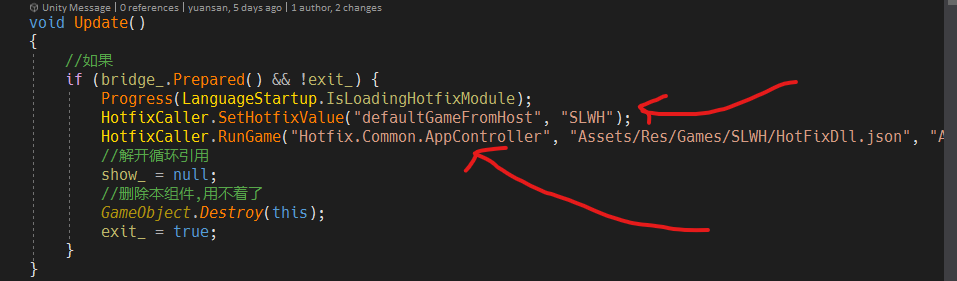
准备工作:

1. 在github下载给定的Unity主工程项目
2. 进入Assets目录,添加git子模块:https://github.com/lenovo734646/ILRuntimeShared.git
3. 进入Hotfix目录,添加git子模块:https://github.com/lenovo734646/ILHotfixCommon.git
4. 进入Hotfix/Games目录,添加git子模块:https://github.com/lenovo734646/Game${ThisGameName}Hotfix.git 并把本地目录名改为${ThisGameName}.

Unity工程中工做的事情:

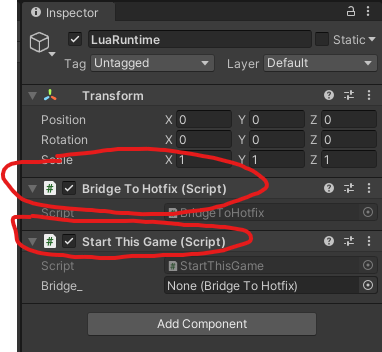
1. 子游戏资源需要放在Assets/Res/Games/${ThisGameName}/ 下.以免Addressables的Key冲突.
2. 子游戏有两个git 子模块,分别在以下目录:
   1. Assets/ILRuntimeShared 这个目录下的代码是unity工程使用的公共库.
   2. Hotfix/ILHotfixCommon 这个目录下的代码是本游戏热更代码公共库.

这两个模块所有子游戏都会使用到.

1. 热更代码的入口为:StartThisGame.cs,把这个脚本挂到一个启动场景上.  
   

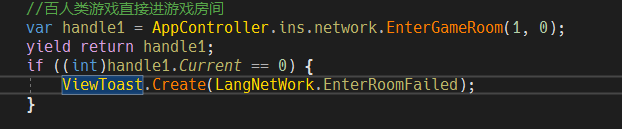
并修改相应的启动数据.其中的SLWH的值修改为ThisGameName 代表的名字

1. 把BridgeToHotfix.cs脚本挂在和StartThisGame.cs相同的GameObject上.



热更工程中要注意的事项:

1. 修改Games/${ThisGameName}/下的.cs文件里的命名空间,改为:Hotfix.${ThisGameName}
2. AppController.cs是热更工程代码的入口.入口函数为:AppController.Start();
3. 本子游戏的入口为:MyApp.Start()
4. 在GameController.cs中处理3个重要回调,这里回调都是子协程:
   1. public virtual *IEnumerator* OnGameLoginSucc(); 这个回调在框架登录子游戏服务器成功时回调,做好相应的界面显示,这个函数完成之后,如果是需要自动进入游戏房间的游戏,客户端调用如下代码进入游戏:



* 1. public virtual *IEnumerator* OnPrepareGameRoom(); 这个回调在框架已锁定了服务器房间座位,要求客户端加载房间资源.这个函数完成之后框架会请求坐入位置.
  2. public virtual *IEnumerator* OnGameRoomSucc();这个回调在框架已完成了进入房间流程时回调.

1. 所有涉及到文字显示的地方,请定义词条在相应的Language.cs里,不可写死在代码里.
2. 配置类的数据,定义在ThisGameConfig.cs里
3. 代码风格保持一致. 非public 成员变量,以\_结尾.
4. 资源加载必须使用以下3个类的成员的方法加载:
   1. AppController.LoadAssets<T> 这个方法加载的资源生存周期为整个APP.
   2. MyApp.LoadAssets<T> 这个方法加载的资源的生存周期为本小游戏.
   3. ViewBase.LoadAssets<T> 这具方法加载的资源生存周期为本界面.

10.Model.cs脚本包含了游戏对像数据的一些定义,公共数据有需要的可以添加在这里.