AssetBundle

整体逻辑：

AssetBundleInit：封装对AssetBundle初始化，热更新等逻辑。

ResourceLoader：对资源加载接口的封装，游戏内想创建资源均从ResourceLoader.Instance.Load() 这个接口获取

AssetBundleManager：AssetBundle管理类，控制着assetBundle的初始化，更新，校验，创建和销毁。

AssetBundleDownload：下载assetbundle的工具类

AssetBundleEditor：AssetBundle相关的editor工具类

AssetBundleOptions：AssetBundle初始化时的配置。

进入游戏的流程：

1. 进入游戏时先通过调用AssetBundleInit.LocalResInit() 方法生成AssetBundleOptions配置文件，再通过调用ResourceLoader.Instance.Init() 方法初始化资源语言和new一个AssetBundleManager来初始化缓存在StreamingAssets路径下的本地资源。此时就可以加载本地资源了（比如loading界面）
2. 然后调用AssetBundleInit.RemoteResInit() 开始热更新检测逻辑。先通过AssetBundleManager. InitRemoteRes() 方法检测网络链接和初始化资源服务器地址。
3. 接着通过AssetBundleManager.DownloadBundleConfig() 方法下载资源服务器上面的bundle配置表，下载完后通过AssetBundleManager.BundleCheck() 方法与本地的bundle配置表进行对比，找出需要更新的bundle资源信息。
4. 再然后通过AssetBundleManager.DownloadBundles() 方法下载需要更新的bundle资源。每下载完一个新的bundle资源都会跟bundle配置表上的md5值进行比对，如果不一样，则重新下载，最多重复3次。每下载成功一个bundle资源后都通过AssetBundleManager.DownloadBundleCallback() 方法缓存到本地。
5. 如果中途某个资源下载失败超过三次，则会通过remoteResInitCallback事件通知外界资源下载失败了，外界还可通过调用AssetBundleManager.ReInitRemoteRes() 方法重新下载
6. 所有需要更新的bundle都下载完毕后就会将最新的bundle配置表缓存到本地，然后通过AssetBundleManager.bundleDownloadFinish() 方法里调用remoteResInitCallback事件告诉外界，所有的资源更新完毕。后面便可随意加载资源了。

Bundle卸载逻辑

从AssetBundleManager里创建的AssetBundle都有着属于自己的计时器（由CreateBundle() 接口创建的AssetBundle除外），游戏里，每隔bundleUnloadInterval（10）秒种会检测一次，如果发现有AssetBundle已超过unloadTime（默认180）秒没有再被使用到，则会通过AssetBundle.Unload(false) 方法将其卸载。如果unloadTime小于0，则永远不卸载。

注意游戏里FairyGui需要的AssetBundle是由AssetBundleManager.CreateBundle() 接口创建的，这些AssetBundle的卸载逻辑由FairyGui内部处理，AssetBundleManager只负责创建。

AssetBundleManager里每隔resourcesUnloadInterval秒还会调用Resources.UnloadUnusedAssets()方法回收内存，但现该功能已屏蔽。

AssetBundle生成和使用规则：

1. 目前 帝国的AssetBundle的资源均在AssetBundles/DayZ目录下，红警的AssetBundle的资源均在AssetBundles/RedAlert目录下，红警的资源不一定是全的，如果代码在RedAlert目录下没有找到AssetBundle资源，则会使用帝国里对应的AssetBundle资源。
2. 把哪些资源合起来生成一个AssetBundle的规则：当一个文件夹下只有文件，没有文件夹的时候，这个文件夹下的所有文件就合起来生成一个AssetBundle。比如AssetBundles/DayZ/PlatformDefault/Default/Prefab/BloodShardEffect这个目录里，只有文件，没有文件夹了，则会将BloodShardEffect文件里所有的资源合并成一个AssetBundle。 因此要注意的是，在Assets/AssetBundles路径下，不能有文件和文件夹同时在一个目录下出现，否则会加载不到对应的资源。
3. AssetBundle的依赖资源大部分都在ResDepends和ResDepends\_RedAlert目录下，但如果某个依赖资源被多个AssetBundle用到，就应该将依赖资源在Assets/AssetBundles里找个目录放好，新生成个AssetBundle。
4. AssetBundle资源有多语言逻辑，加载资源时会先找对应语言目录下的AssetBundle资源，如果没有找到AssetBundle资源或者在AssetBundle里没加载到想要的资源，则会去Default目录下加载资源。
5. IOS资源与上面的第4条类似，如果是ios包，则会优先加载IOS目录下的AssetBundle，如果没有加载到则会去PlatformDefault目录下加载。

AssetBundle相关工具：

1. 在Resources目录下，有个叫GameConfigs的游戏配置文件，里面有个类AssetBundleSwitch，有一些关于AssetBundle的控制开关。
2. isUseEditorRes：直接使用AssetBundles目录下的资源，游戏开发时方便使用。如果不勾此选项，则会走正式逻辑，从资源服务器下载bundle配置表和AssetBundle资源，然后加载更新好的和StreamingAssets已有的AssetBundle资源进入游戏。
3. simulateLocalServer：使用本地资源服务器，如果需要测试更新逻辑，可勾选此选项，并且勾选上unity上方的工具栏里的"项目工具/AssetBundle相关工具/Local AssetBundle Server" 来开启本地服务器，这样后续进入游戏时就会从本电脑的"SevenKingdom\Master\UnityProject\AssetBundles\" 的路径下下载资源而不是从资源服务器上下载了。
4. url：资源服务器地址。
5. resourcesVersion：客户端自定义的资源版本号
6. minDownloadSizeDialog：更新资源时，超过多少M就需要弹窗提示玩家需要更新资源
7. unity 上方的工具栏里 "项目工具/AssetBundle相关工具/" ，这里面还有些其他的有关AssetBundle的一些工具。每个工具的含义基本上都是上面的字面意思。

注意的几个点

1. 目前打包时会将所有的多语言资源都会放到包里，但是在更新的时候，只会下载对应语言的资源，并不会将所有的多语言资源下载下来。
2. 新包上线后AssetBundle处理流程。当新包上线后，需要将新上线包的AssetBundle配置表备份到svn目录<https://10.80.9.41/svn/SevenKingdom/Master/AssetBundleServer>下这样下次为上线的包打新的AssetBundle资源的时候，打AssetBundle的工具会自动与上次备份的配置表进行比对（需要提前在AssetBundleEditor. assetBundleSeverPath字段里提前配置好备份配置表的路径），然后就会将更改了的AssetBundle资源放置在AssetBundleServer里的change目录下。将该目录下修改了的AssetBundle资源上传，由服务器更新，将其上传到资源服务器。