

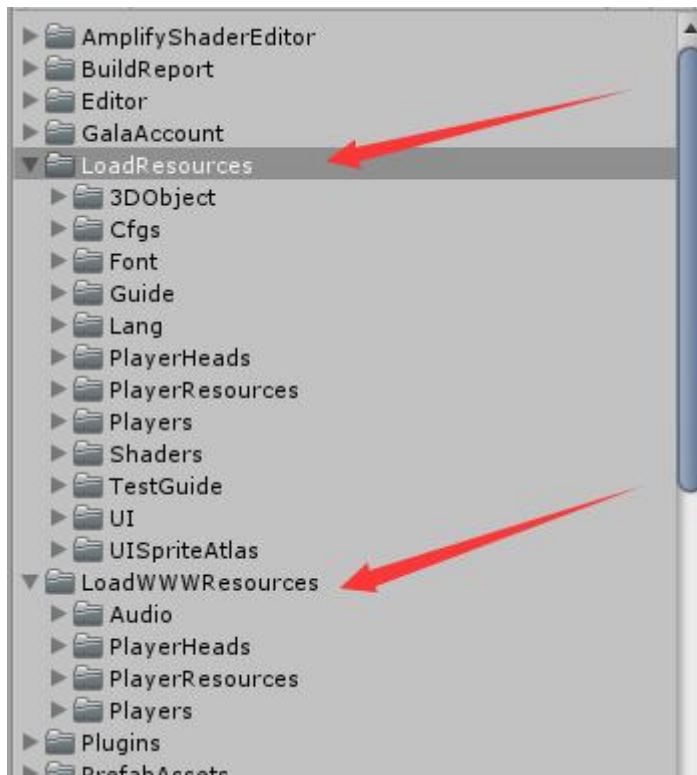
优点

- A、减小安装包大小
- B、更好的控制资源的内存占用
- C、更新资源
- D、减少打安装包的时间

缺点

- A、逻辑复杂度
- B、增加下载功能
- C、资源加载逻辑
- D、资源卸载逻辑
- E、加载时序问题解决
- F、添加生成 AssetBundle 逻辑

1. 构建 AssetBundle



1>.LoadResource 目录下的资源会被打成 assetbundle 放到本地（StreamingAssets 目录下），该目录下后续更新资源可以通过 patch 更新

LoadResource 目录下的资源可以用同步方式读取，也可以使用异步

2>LoadWWWResources 目录下的资源会被打成 assetbundle，放在 cdn 上，运行时根据需要进行加载
LoadWWWResources 目录下的资源只能通过异步方式读取

3>

D:) > work_space > soccer_client > AssetBundle_Server > CurrentVersion_1.0.0 > Android >				
名称	修改日期	类型	大小	
OffLine_Test	2019/7/10 2:06	文件夹		
OnLine	2019/7/10 1:26	文件夹		
OnLine_Test	2019/7/10 13:57	文件夹		

> soccer_client > AssetBundle_Server > CurrentVersion_1.0.0 > Android > OnLine >				
名称	修改日期	类型	大小	
CacheAssetBundles	2019/7/10 1:20	文件夹		
LocalAssetBundles	2019/7/10 1:40	文件夹		
PatchPkg	2019/7/10 1:40	文件夹		

> AssetBundle_Server > CurrentVersion_1.0.0 > Android > OnLine > LocalAssetBundles				
名称	修改日期	类型	大小	
00b2a7c1b1e330c4a1b8d0d6dec5f5d8	2019/7/10 1:07	文件	24 KB	
00b2a7c1b1e330c4a1b8d0d6dec5f5...	2019/7/10 1:07	MANIFEST 文件	1 KB	
00b18a05ce0246ad6903f7891148f9ac	2019/7/10 1:06	文件	14 KB	
00b18a05ce0246ad6903f7891148f9a...	2019/7/10 1:06	MANIFEST 文件	2 KB	
00b648d4610e4a11a49de45854569f6b	2019/7/10 1:09	文件	4 KB	
00b648d4610e4a11a49de45854569f6...	2019/7/10 1:09	MANIFEST 文件	1 KB	
00c3d4a095de6ccb390e080ad9b2f95d	2019/7/10 1:16	文件	269 KB	
00c3d4a095de6ccb390e080ad9b2f95...	2019/7/10 1:16	MANIFEST 文件	1 KB	
00d17405b0525f5a5b27034c70d7afaf	2019/7/10 1:10	文件	1,673 KB	
00d17405b0525f5a5b27034c70d7afa...	2019/7/10 1:10	MANIFEST 文件	4 KB	
00e27b61f783ba237bd042ec4ee02a17	2019/7/10 1:15	文件	4 KB	
00e27b61f783ba237bd042ec4ee02a1...	2019/7/10 1:15	MANIFEST 文件	1 KB	
00eca49e6f994457f03cec2cbbd79d05	2019/7/10 1:17	文件	237 KB	
00eca49e6f994457f03cec2cbbd79d0...	2019/7/10 1:17	MANIFEST 文件	1 KB	
0a5ba7f998e64a1e6e633cca99c31eab	2019/7/10 1:14	文件	3 KB	
0a5ba7f998e64a1e6e633cca99c31eab	2019/7/10 1:14	MANIFEST 文件	1 KB	

> AssetBundle_Server > CurrentVersion_1.0.0 > Android > OnLine > PatchPkg >				▼	🔄
名称	修改日期	类型	大小		
CacheAssetBundles	2019/7/10 1:40	文件夹			
Patch_1.0.1.zip	2019/7/10 1:29	好压 ZIP 压缩文件	1,479 KB		
Patch_1.0.2.zip	2019/7/10 1:40	好压 ZIP 压缩文件	1,915 KB		
update_pkg_file.bytes	2019/7/10 1:40	BYTES 文件	1 KB		

```

1 {
2     "VersionNo": "1.0.2",
3     "ChannelAppDownUrl": "www.google.com",
4     "DownloadList": [
5         {
6             "assetName": "Patch_1.0.1.zip",
7             "assetMD5": "a8e213f9c9d369e72df40b06eb5ecd7e",
8             "assetSize": 1513604,
9             "versionNo": "1.0.1",
10            "FileCount": 3
11        },
12        {
13            "assetName": "Patch_1.0.2.zip",
14            "assetMD5": "008ec1066858236e9ea9c40e8859c5de",
15            "assetSize": 1960548,
16            "versionNo": "1.0.2",
17            "FileCount": 3
18        }
19    ]
20 }

```

work_space > soccer_client > Assets > StreamingAssets > AssetBundles > Android

名称

修改日期

类型

00b2a7c1b1e330c4a1b8d0d6dec5f5d8

2019/7/10 2:09

文件

00b2a7c1b1e330c4a1b8d0d6dec5f5d8.meta

2019/7/10 15:39

META 文件

00b18a05ce0246ad6903f7891148f9ac

2019/7/10 2:08

文件

00b18a05ce0246ad6903f7891148f9ac.meta

2019/7/10 15:39

META 文件

00b648d4610e4a11a49de45854569f6b

2019/7/10 2:12

文件

00b648d4610e4a11a49de45854569f6b.meta

2019/7/10 15:39

META 文件

00c3d4a095de6ccb390e080ad9b2f95d

2019/7/10 2:23

文件

00c3d4a095de6ccb390e080ad9b2f95d.meta

2019/7/10 15:40

META 文件

00d17405b0525f5a5b27034c70d7afaf

2019/7/10 2:12

文件

00d17405b0525f5a5b27034c70d7afaf.meta

2019/7/10 15:39

META 文件

00e27b61f783ba237bd042ec4ee02a17

2019/7/10 2:20

文件

00e27b61f783ba237bd042ec4ee02a17.meta

2019/7/10 15:39

META 文件

00eca49e6f994457f03cec2cbbd79d05

2019/7/10 2:23

文件

00eca49e6f994457f03cec2cbbd79d05.meta

类型: 文件

2. 更新下载

进入游戏判断是否需要下载，下载的方式，通过下载 zip 包（支持增量更新），下载完后检查 md5，解压至本地

3. 加载

```
public enum AssetInMemoryType
{
    Normal = 0,          //资源销毁即清除相应内存
    TempResident = 1,    //跨场景时即清除相应内存
    Resident = 2,        //常驻内存
}
```

```
/// <summary>
/// 同步加载
/// </summary>
/// <typeparam name="T">类型</typeparam>
/// <param name="path">路径</param>
/// <param name="attachGo">附加销毁脚本的GameObject</param>
/// <param name="assetInMemoryType">希望在内存中存在的形式</param>
/// <returns>加载结果</returns>
99+ 引用 | -luke-, 11 天前 | 1 名作者, 1 项更改
T LoadSync<T>(string path, GameObject attachGo = null, AssetInMemoryType assetInMemoryType = AssetInMemoryType.Normal) where T : Object;
```

使用方式:

```
var itemRes = GameKernel.ResMgr.LoadSync<GameObject>("UI/PlayerDetail/Item/PlayerGroupItem");
```

```
/// <summary>
/// 同步加载
/// </summary>
/// <typeparam name="bp">加载参数</typeparam>
/// <returns>加载结果</returns>
2 个引用 | -luke-, 2 天前 | 1 名作者, 1 项更改
Object LoadSync(BundleParams bp);
```

```
/// <summary>
/// 同步加载
/// </summary>
/// <param name="path">路径</param>
/// <param name="parent">父节点</param>
/// <param name="assetInMemoryType">希望在内存中存在的形式</param>
/// <returns>Instantiate后的GameObject</returns>
99+ 引用 | -luke-, 36 天前 | 1 名作者, 1 项更改
GameObject LoadGoSync(string path, Transform parent = null, AssetInMemoryType assetInMemoryType = AssetInMemoryType.Normal);
```

使用方式:

```
GameObject tabBar = GameKernel.ResMgr.LoadGoSync("UI/Components/TabBar");
```

```
/// <summary>
/// 异步加载
/// </summary>
2 个引用 | -luke-, 11 天前 | 1 名作者, 1 项更改
void LoadAsync<T>(string path, GameObject attachGo, LoadAssetOverHandler handle, AssetInMemoryType assetInMemoryType = AssetInMemoryType.Normal) where T : Object;
```

```
/// <summary>
/// 异步加载
/// </summary>
/// <typeparam name="T">类型</typeparam>
/// <param name="path">路径</param>
/// <param name="handle">加载回调</param>
/// <param name="assetInMemoryType">希望在内存中存在的形式</param>
void LoadAsync<T>(string path, LoadAssetOverHandler handle, AssetInMemoryType assetInMemoryType = AssetInMemoryType.Normal) where T : Object;
```

使用方式:


```
GameKernel.ResMgr.LoadAsync<Sprite>("UI/Dynamic/ActivityBgImg/Event_RechargeFeedbackBg", (info) =>
{
    _bg.sprite = info.MainAsset as Sprite;
    _bg.Alpha(1);
});
```

```
/// <summary>
/// 支持await的异步加载
/// </summary>
/// <typeparam name="T">类型</typeparam>
/// <param name="path">路径</param>
/// <param name="assetInMemoryType">希望在内存中存在的形式</param>
/// <returns>加载结果</returns>
37 个引用 | -luke-, 36 天前 | 1 名作者, 2 项更改
Task<T> LoadAsync<T>(string path, AssetInMemoryType assetInMemoryType = AssetInMemoryType.Normal) where T : Object;
```

使用方式:

```
GameObject obj = await GameKernel.ResMgr.LoadAsync<GameObject>(RecoverPlayerPath);
_playerController = (GameKernel.ResMgr.Instantiate(obj, transform) as GameObject).GetComponent<PlayerLowModelController>();
```

```
/// <summary>
/// 支持await的异步加载
/// </summary>
3 个引用 | -luke-, 11 天前 | 1 名作者, 1 项更改
Task<T> LoadAsync<T>(string path, GameObject attachGo, AssetInMemoryType assetInMemoryType = AssetInMemoryType.Normal) where T : Object;
```

```
/// <summary>
/// 此类接口返回的是Instantiate后的GameObject,此接口加载的资源,被销毁时资源即会被释放
/// 注:此处非常重要,正常情况下,凡是使用GameObject.Instantiate的地方需要使用GameKernel.ResMgr.Instantiate,否则使用不当,
/// 会造成内存泄漏
/// </summary>
/// <param name="obj">Object</param>
/// <param name="parent"></param>
/// <returns>实例化后的Object</returns>
99+ 引用 | -luke-, 28 天前 | 1 名作者, 3 项更改
GameObject Instantiate(Object obj, Transform parent = null);
```

使用方式:

```
GameObject pop = GameKernel.ResMgr.Instantiate(item);
```

4. 卸载

- (1) GameObject 销毁引用计数自动减 1, 每个 AssetBundle 的引用计数为 0, 即马上会被卸载
- (2) 通过调用 ResourceMgr 中的函数进行卸载

```
/// <summary>
/// 卸载资源
/// </summary>
/// <param name="path">资源路径</param>
/// <param name="type">资源类型</param>
/// <returns>卸载是否成功</returns>
bool UnloadResource(string path, System.Type type);
```

```

/// <summary>
/// 卸载其他场景AB
/// </summary>
/// <param name="sceneName">场景名</param>
/// <param name="type">资源在内存中类型，常驻/临时常驻/普通资源</param>
3 个引用 | -luke-, 41 天前 | 1 名作者, 1 项更改
void UnloadOtherSceneAB(string sceneName, AssetInMemoryType type);|

```

path 路径与加载路径一致

type 与加载资源类型一致

5. 编辑器灵活使用资源加载

