

热更新使用文档

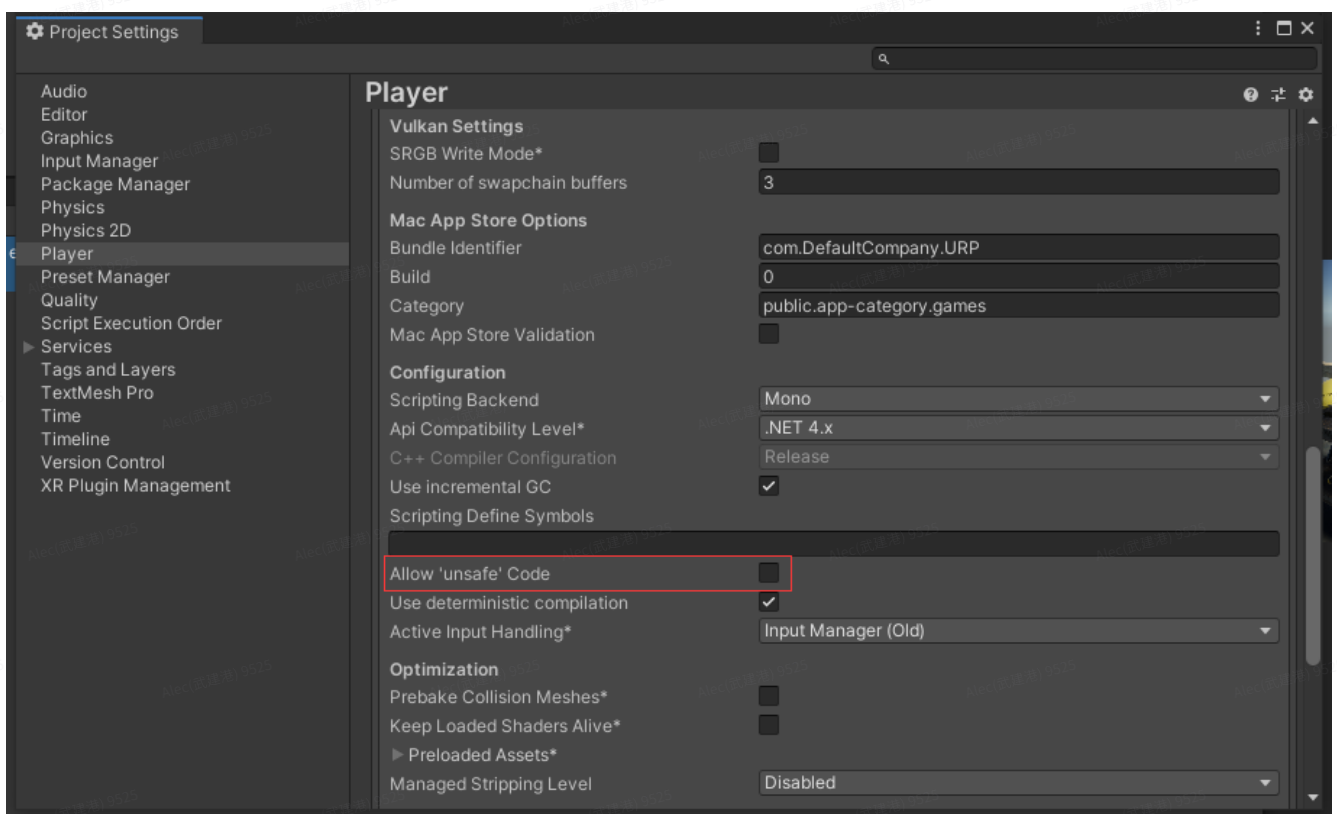
插件地址：

<https://git.devcloud.ztgame.com/engine/tech/hotupdate>

1. 导入插件。

导入后如果出现：Unsafe code may only appear if compiling with /unsafe.报错

打开工具栏 “Edit/Project Settings/Player/Other Settings/Allow 'unsafe' Code” 选项。



2. 在本地或者云服务器上建立WEB文件服务器。

3. 打包设置：

设置对应的配置文件服务器地址：

GUpdate.CS

```
//用于判定这个包里是否激活了GUpdate
public static bool IsGupdate = false;

//用于测试的配置文件
private string testConfig = Application.persistentDataPath + "/gupdate.json";
//Android下载测试文件的地址
private string downloadConfigAndroidUrl = "http://192.168.84.179:8888/GUpdateTest/gupdate.json";
```

```
private string downloadConfigAndroidUrl = "http://192.168.94.178:8080/UpdateTest/gupdate.json";  
//ios下载测试文件的地址  
private string downloadConfigIosUrl = "http://192.168.94.178:8080/UpdateTest/gupdate.json";
```

关闭处理Wwise音效。(如需要使用Wwise，后续可以打开并阅读相关代码)：

GUpdateBuilder.CS

```
///   
//是否处理Wwise音效  
private static bool HandleWwiseFile = false;
```

配置文件设置：gupdate.json

C#

```
1 {  
2     "host": "http://192.168.x.x:8080",  
3     "checkmd5": true,  
4     "release": false,  
5     "whole": true,  
6     "version": "1.020.4",  
7     "channel": "UpdateTest/versions"  
8 }
```

host:资源远程路径，如果本地手机测试请写192.168.x.x局域网地址

checkmd5:下载好资源是否md5校验，在某些场景下可能有用。。建议打开

release:为true时，不会弹出版本选择面板，所以无法版本回退，直接更新到服务器最高版本，并且取消后无法进入游戏，一般开发填false

whole:是否为整包，分包模式安装完apk后会强制同步一次服务器版本

version:apk版本号，不用手动填写，发布时自动填写

channel:对应发布的平台，根据平台会存储于不同的平台目录，加载不同的资源。

之后根据平台来设置选用的地址。

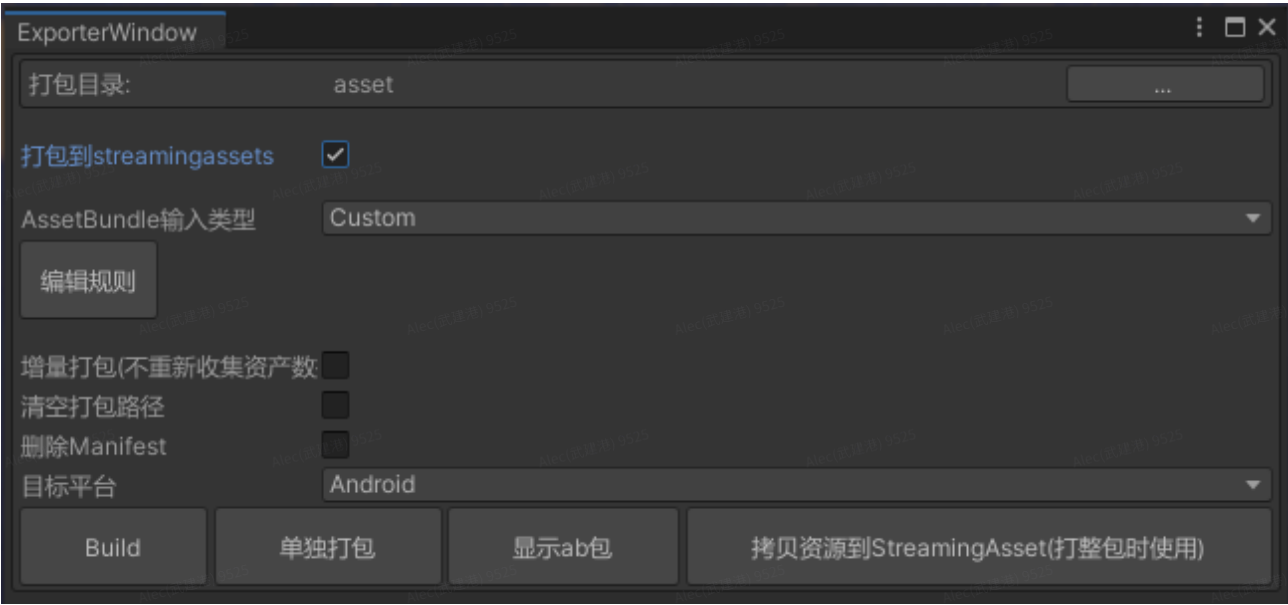
修改脚本为：GUpdate.cs

C#

```
1  IEnumerator PrearLoadConfig()
2  {
3      Debug.Log("PrearLoadConfig");
4      if (!File.Exists(testConfig))
5      {
6          //主要是这些修改: : :
7          //-----
8
9          //用于动态调整配置做内外网测试
10         string DefiniteUrl = "";
11
12         if (m_platform == "android")
13         {
14             DefiniteUrl = downloadConfigAndroidUrl;
15         }
16         else if (m_platform == "ios")
17         {
18             DefiniteUrl = downloadConfigIosUrl;
19         }
20         else
21         {
22             Debug.Log("Not Support This Platform!!!");
23         }
24         //-----
25
26
27         Debug.Log("DefiniteUrl:::" + DefiniteUrl);
28         UnityWebRequest www = UnityWebRequest.Get(DefiniteUrl);
29         www.timeout = 2;
30         yield return www.SendWebRequest();
31         if (www.isNetworkError || www.isHttpError)
32         {
33             Debug.Log("download test config failed");
34         }
35         else
36         {
37             string txt = www.downloadHandler.text;
38             File.WriteAllText(testConfig, txt);
39             Debug.Log("download test config success");
40         }
41     }
42
43     m_behaviour.StartCoroutine(LoadConfig());
44 }
```

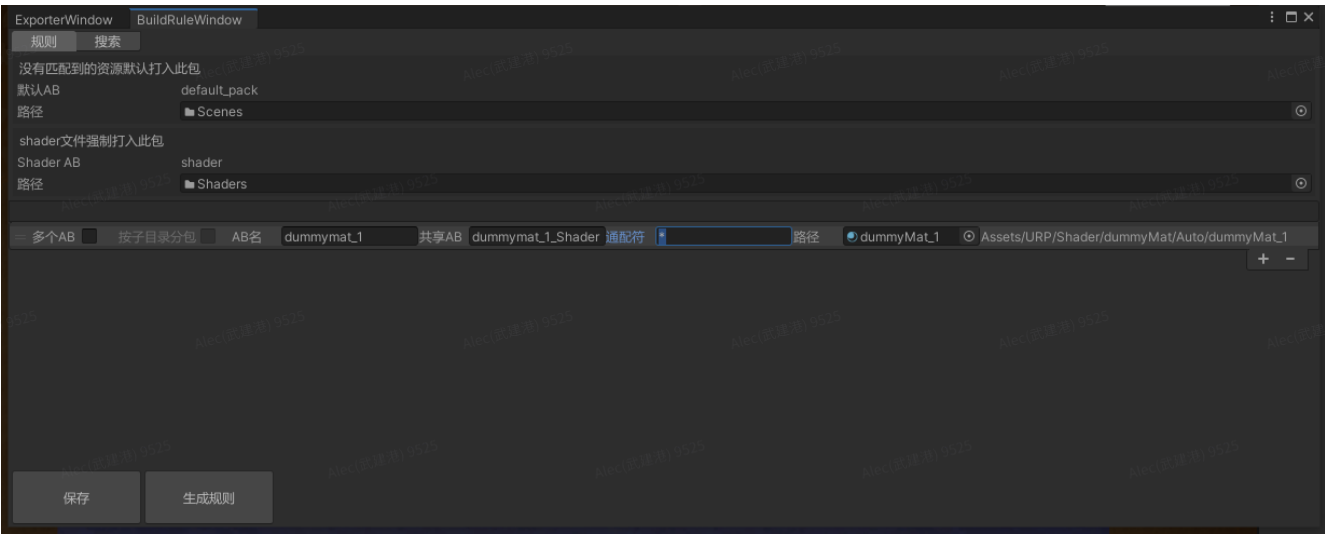
4. 打包资源：

打开Untiy工具栏的Giant Tools/AssetBundles/Show AssetBundles(打包资源).



打包类型选择Custom。

点击编辑规则：



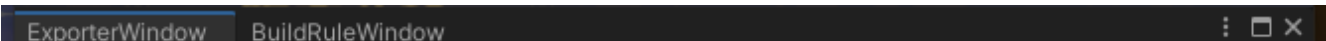
默认AB路径，选择资源的主目录，未分配指定AB包的资源都会打包在default_pack包内。

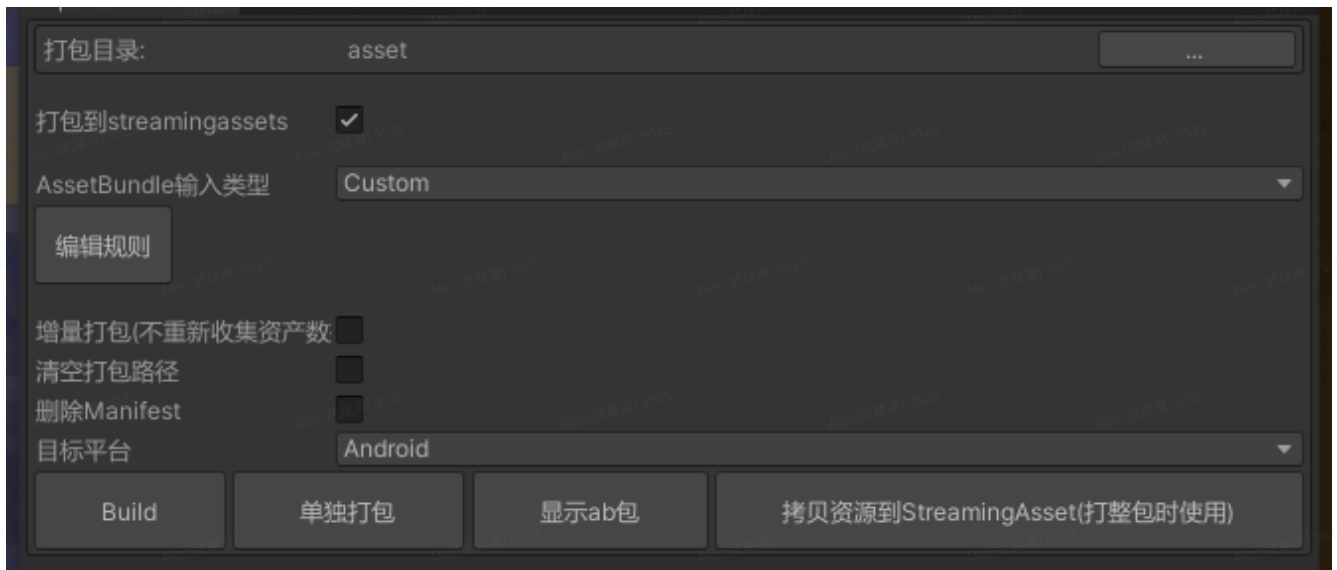
ShaderAB路径，选择Shader目录。

后续点击下方+号可以添加具体AB包打包规则，可以设置所有资源的AB包打包分配，可以选择非默认AB路径下的文件夹，也可以分配默认AB包的资源。

之后点击生成规则，点击保存。

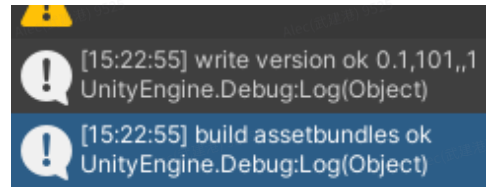
回到之前的打包界面：



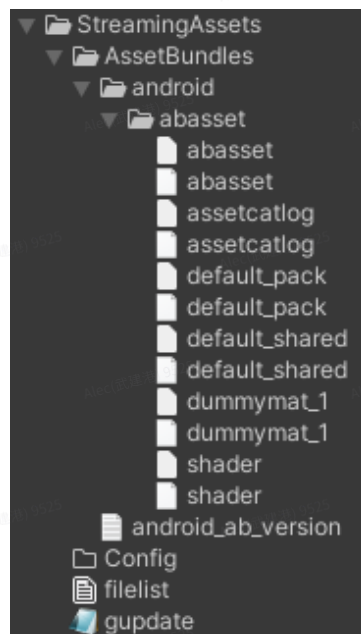


选择目标平台，点击Build。

看到Log表示打包成功：

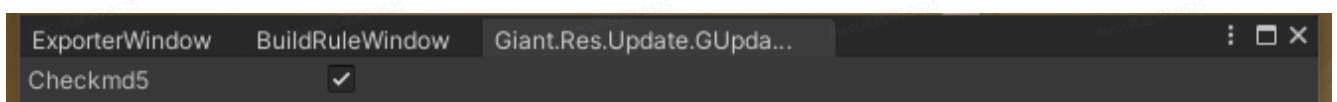


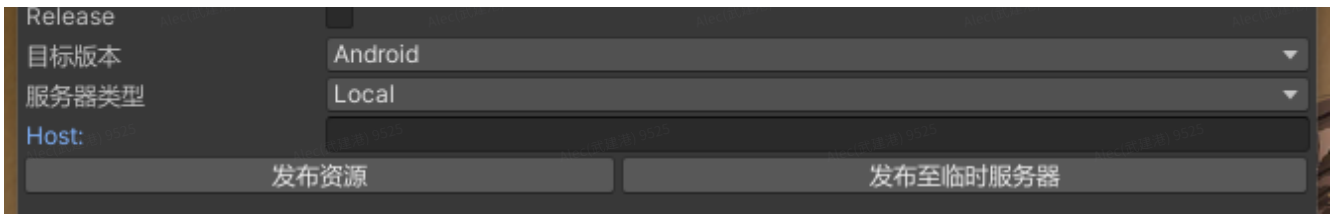
打包后的资源路径为：



5. 发布资源：

点开Unity工具栏的Tools/GUUpdate/OpenWindow。





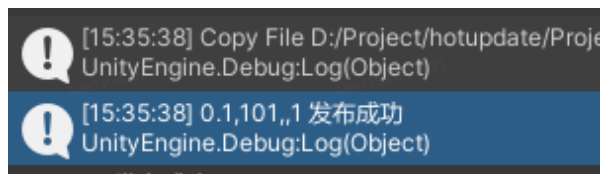
自动发布资源：

需要配置CI流水线，这部分暂时待完善。

手动发布资源：

选择号目标版本，服务器类型选Local，不需要填Host.直接点发布资源。

看到Log表示发布成功：



发布的资源会被copy到项目目录的同级：



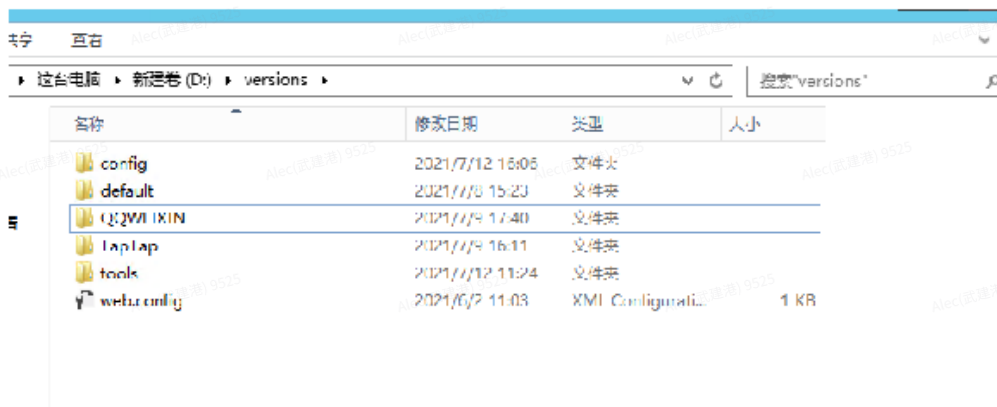
放到WEB段对应目录，目录结构如下文档：

Gupdate CDN 文件布局 and 工具链

简介: 主要是介绍放置大陆 CDN 上热更资源的的文件布局, 以及对应的 CICD

根目录:

config 存一些配置信息, tools 目录是一些工具脚本, 其他文件夹都是渠道名字

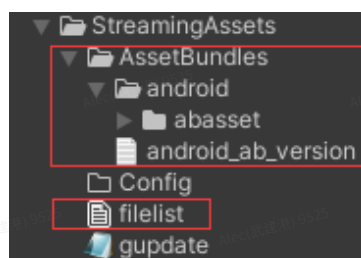


名称	修改日期	类型	大小
config	2021/7/12 16:05	文件夹	
default	2021/7/8 15:23	文件夹	
QQWIKIN	2021/7/9 17:40	文件夹	
Laptop	2021/7/9 16:11	文件夹	
tools	2021/7/12 11:24	文件夹	
webui.config	2021/6/2 11:03	XML Configuration	1 KB

Gupdate CDN文件布局 and 工具链.docx

由于是发布Local模式, 本机的Assets/StreamingAssets/filelist 会是本次打包的热更新文件数据。

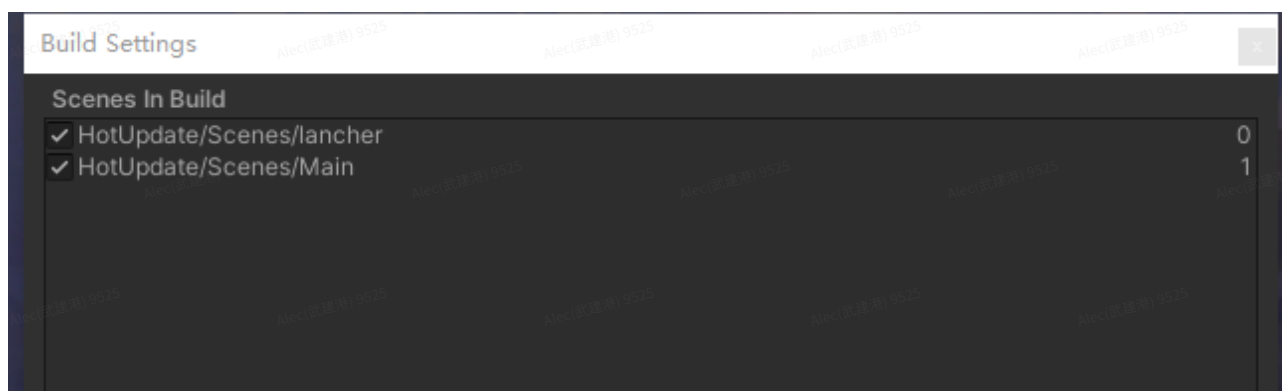
如果需要测试热更新本次打包的资源, 可以删除项目下的AssetBundles目录, 清空filelist文本:

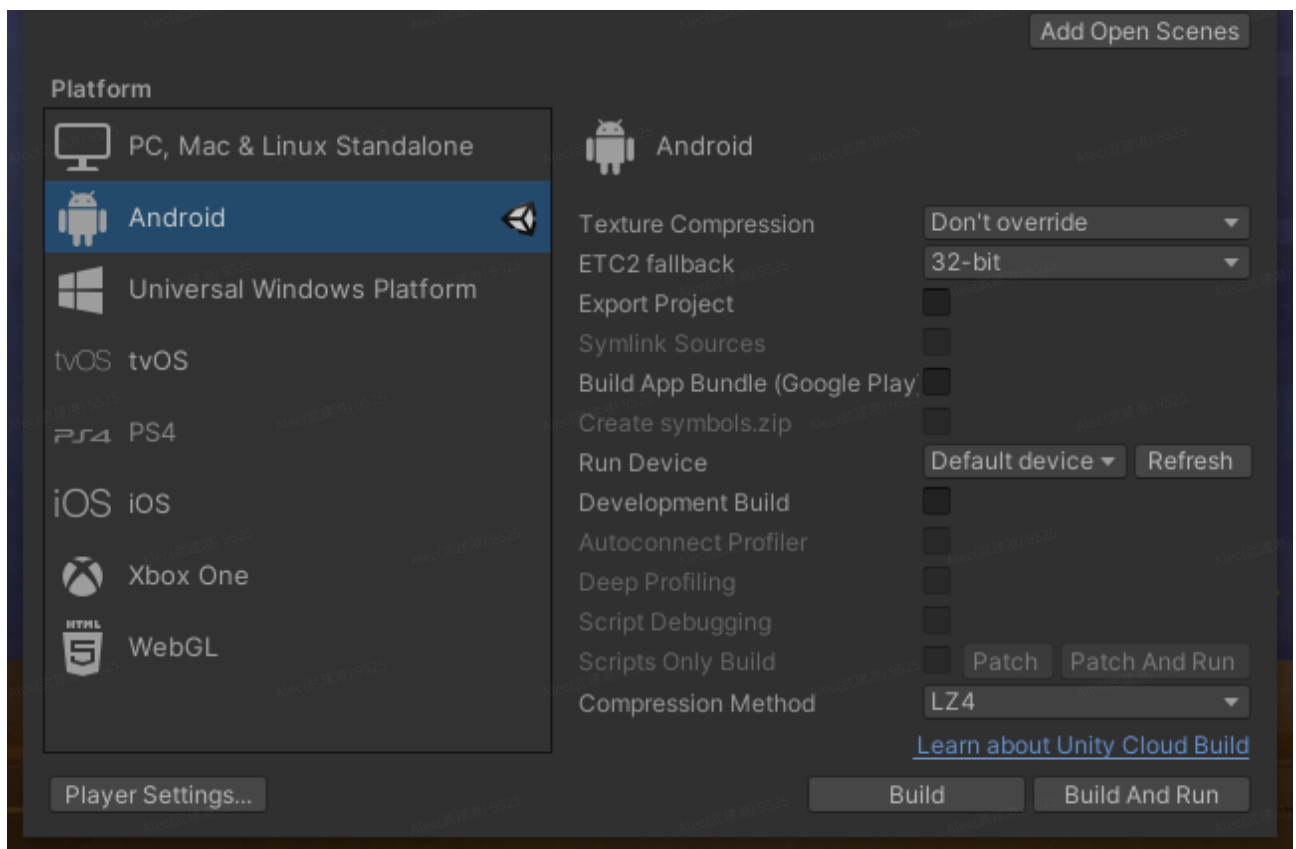


提示: 如果路径有问题, 可以根据报错来调整路径。

6. 打包项目APK/IPA测试

Build Settings





只设置此两个目录，直接打包，其他场景打包到AB包内，热更下来后加载。
运行：



安装包版本号: 1.020.4
资源包版本号: 1,020,6

会显示出所有热更新的版本资源目录（之前配置文件gupdate.json release 为false则直接不显示此面板）。

点击更新后，会进入更新界面，更新完了进入Main场景。

具体流程请看GUpdater.cs。

7. 资源加载流程

场景加载：

C#

```
1 private string loadScenesName;
2 // private xxxx reg...
3
4 private void Start()
5 {
6     StartCoroutine(LoadScene("Assets/Scenes/SampleScene"));
7 }
8
9 /// <summary>
10 /// 管理加载和卸载场景
11 /// </summary>
12 /// <param name="ScenePath">场景路径</param>
13 /// <returns></returns>
14 IEnumerator LoadScene(string ScenePath)
15 {
16     //判断之前是否加载场景，已经加载，则卸载。
17     if (!string.IsNullOrEmpty(loadScenesName))
18     {
19         req = Giant.Res.AssetsMgr.UnloadSceneAsync(loadScenesName);
20         while (!req.IsCompleted)
21             yield return null;
22     }
23
24     loadScenesName = ScenePath;
25
26     req = Giant.Res.AssetsMgr.LoadSceneAsync(loadScenesName);
27
28     while (!req.IsCompleted)
29         yield return null;
30 }
```

预设加载：

C#

```
1 private void Start()
2 {
3     //加载预设，传入实例化的坐标。
4     var req = Giant.Res.AssetsMgr.InstantiateGameObject(
5         "Assets/Scenes/Cube.prefab",
6         Vector3.zero, Quaternion.Euler(Vector3.zero));
7 }
```

其他资源加载：

C#

```
1 private void Start()  
2 {  
3     //加载资源，传入目标资源类型。  
4     Giant.Res.AssetHandle assetHandle = Giant.Res.AssetsMgr.LoadAssetHandle(  
5         "Assets/Scenes/Test2Material.mat",  
6         typeof(Material));  
7  
8     //资源实例化。  
9     if (null != assetHandle)  
10    {  
11        GameObject.Find("Cube").GetComponent<MeshRenderer>().material  
12        = assetHandle.Instantiate<Material>();  
13    }  
14 }
```