# 软件部门前端开发规范化文档

V 1.0

**章节目录**

[软件部门前端开发规范化文档 1](#_Toc943463282)

[一、 项目版本递交规范 2](#_Toc1915614773)

[1. 打包流程及文件夹命名 2](#_Toc662653987)

[ Demo版本打包 2](#_Toc375366167)

[ IGT安装版本打包 2](#_Toc1619697530)

[ IGT打包主工程源码包 3](#_Toc753677338)

[ IGT打包Fuel Package 4](#_Toc1196469760)

[ IGT打包UCGS 5](#_Toc30385812)

[2. 版本修复记录的格式及递交说明 6](#_Toc1740717945)

[ 修复记录的格式 6](#_Toc1076778534)

[ 版本递交说明 6](#_Toc572127669)

[3. 版本正式递交内部审核规范 6](#_Toc1465445264)

[ Demo及公司内部项目 6](#_Toc248604605)

[ IGT递交PA Test之前 7](#_Toc1441902820)

[ IGT正式递交PA 9](#_Toc1855222992)

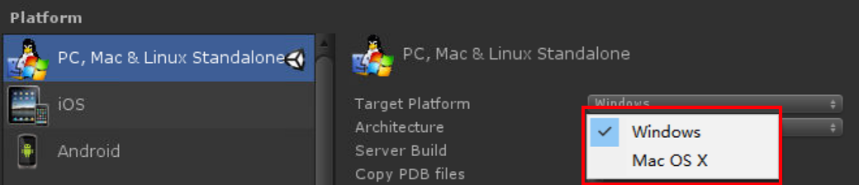
[ IGT递交PA Test之后 10](#_Toc1417755751)

[附件： 12](#_Toc1889843592)

1. **项目版本递交规范**
2. **打包流程及文件夹命名**

* **Demo版本打包**

先选择对应平台（Windows或Mac OS X），然后点击Build按钮并创建文件夹打包。



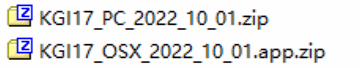
文件夹名称由“项目名+编号\_平台信息\_日期”三个部分组成（例如：XXX17\_OSX\_2022\_10\_01、XXX04\_PC\_2022\_10\_01),月份和日期保持两位数，字母全部大写，保持格式工整，养成好习惯！

打包Mac版本需要注意，项目打包后文件夹名称会有变化（在原有的基础上增加了“.app”后缀名），切记不可删除。



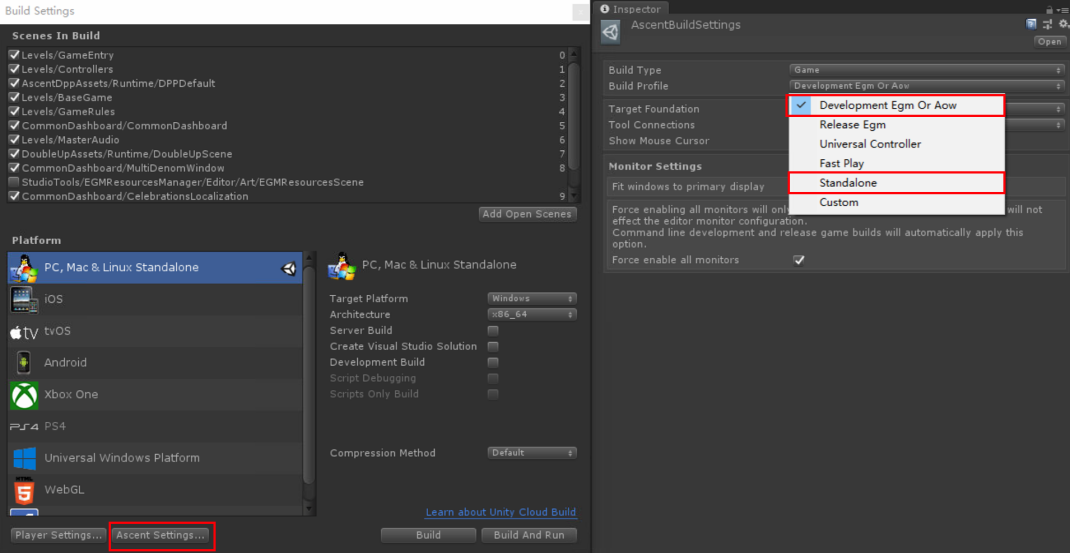
打包完成后如果是PC版本需要先运行一下游戏，确保游戏不会出现闪退、崩溃、无法打开等问题，若OSX版本有条件的情况下也需要进行基本测试。

最后，无论是PC还是OSX版本，将打包后的文件夹直接压缩成Zip格式（**注意：**必须是Zip格式，尤其是OSX版本，用其他压缩格式会出问题）打包后文件如下：

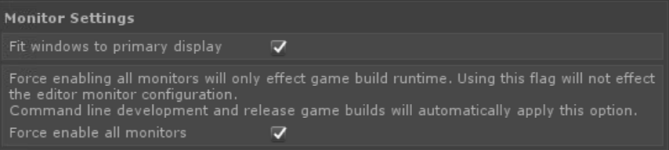


* **IGT安装版本打包**

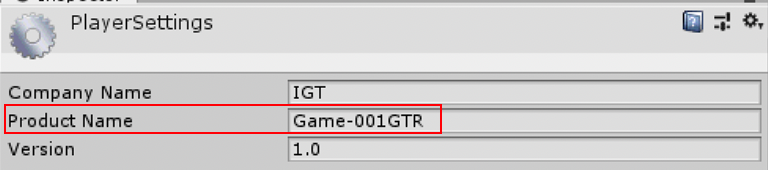
Platform固定选择Windows,然后先检查Ascent Build Settings设置，选择好发布目标（Development Egm Or Aow或Standalone，**注意：**EGM对应游戏机平台，Standalone对应PC平台）。



如果是打包Standalone版本需要勾选以下两个选项，确保屏幕适配正确（可以一直保持勾选）。



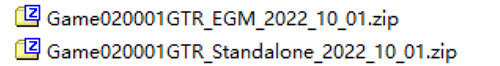
第二步，打开Player Settings，确保Company Name和Product Name填写正确，IGT每个游戏的ProductName都是一个独立编号，所以，第一次打包时需要确认。



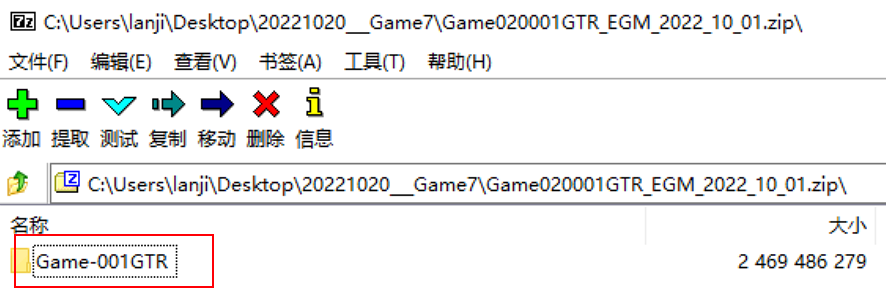
第三步，创建和Product Name相同名称的文件夹，点击Build按钮进行打包。

第四步，打包好之后需要到对应平台中运行游戏，防止出现闪退、崩溃、红屏、无法运行等问题。

最后，将打包好的文件夹直接压缩成Zip格式（同样必须是Zip格式，客户要求），名称由“项目编号\_类型\_日期”三个部分组成，Product Name和Player Settings中的设置保持一致，Game大小写必须统一，保持格式工整。最终文件如下：

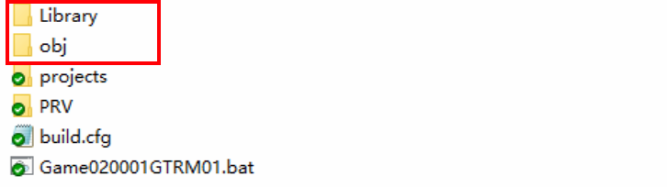


压缩包内应该只包含一个和Product Name同名的文件夹，不包含其他任何东西并且不允许多层嵌套。

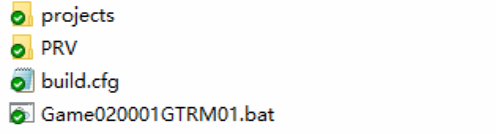


* **IGT打包主****工程源码包**

首先检查工程Assets同级目录下的AssemblyProjects文件夹是否包含AssemblyProjects\ExtensionServices\ExtensionServices.AssemblyConfig文件，如果有的话需要删除，然后关闭Unity,将工程里的Library和Obj文件夹移动到项目之外（主要目的是为了防止被包含，不需要删除，打完包还需要移动回来）。



然后将根目录的projects、PRV文件夹和“.cfg”、“.bat”文件共计四部分内容直接打包成Zip格式（不允许使用其他压缩格式）如下所示：



其中，projects文件夹内包含Unity工程。

PRV文件夹内包含一份记录Defect修复的ChangeList文件，如果没有需要自行创建（命名为“[Product Name]\_Defect\_RepairRecord”），在正式递交PA之前，该文件夹可以不包含。

“.cfg”和“.bat”一个是配置文件一个是用于自动化构建的脚本因此必须被包含，其中“.bat”文件的名称必须是完整的项目编号，即：“[Product Name] + M01”。

打包成Zip之后（不允许使用其他压缩格式）将文件命名为“项目编号\_类型\_日期”格式，如下所示:



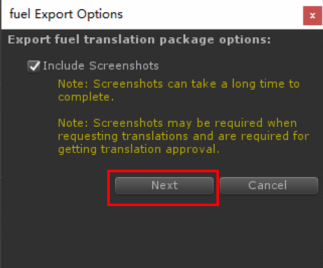
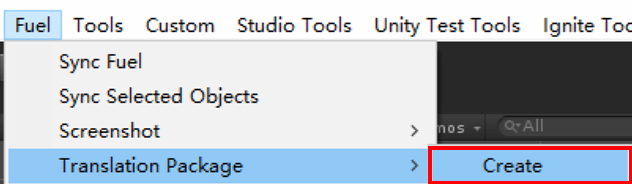
压缩包内不应该包含任何其他嵌套的文件夹或中文相关内容，结果如下所示：



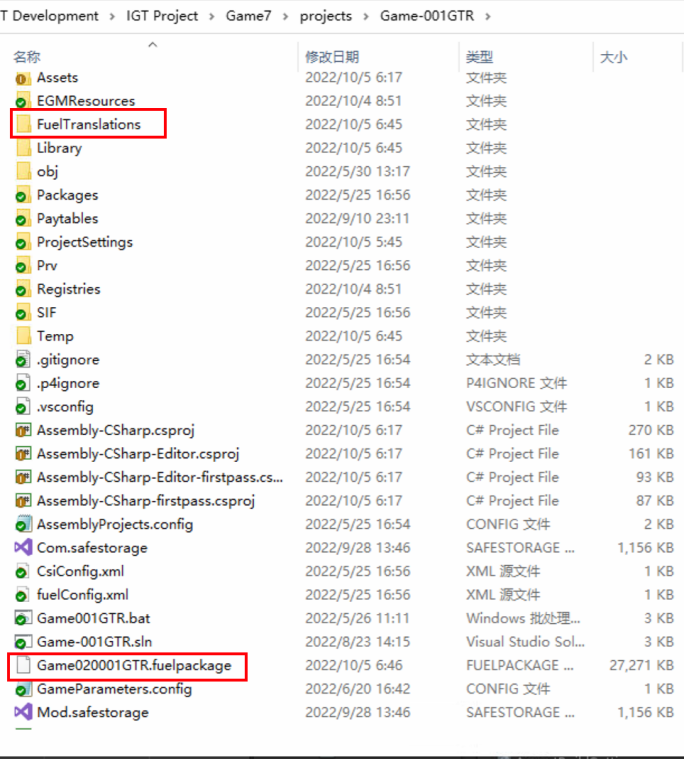
* **IGT打包Fuel Package**

要注意先在Unity中将需要翻译的内容设置好，关于这部分如何设置的问题可以参考“[Fuel Localization Process for Third Parties.pdf](#Fuel_Localization_Process_For_Third)”文件。

第一步，在Unity的菜单栏点击Fuel->Translation Package->Create选项，在弹出框中点击Next按钮即可进行导出操作。

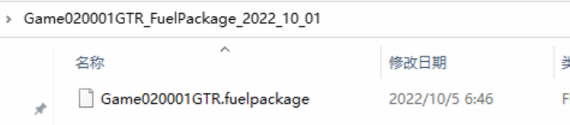


第二步，在项目的Assets同级目录中找到“.fuelpackage”文件，该文件包含了所有需要翻译的信息，另外还有一个FuelTranslations文件夹，这个文件夹包含了当前工程内所有的文字图像截图和动画视频，可以先删掉，防止项目冗余。

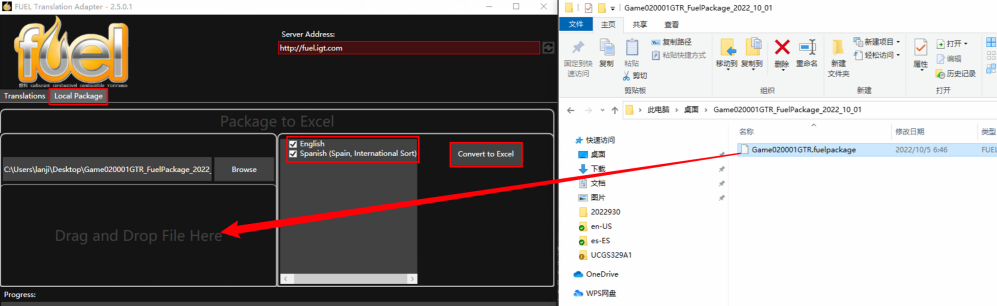


第三步，在桌面新建一个文件夹，命名格式为“[Product Name]\_类型\_日期”并将“.fuelpackage”文件移动到该文件夹内。

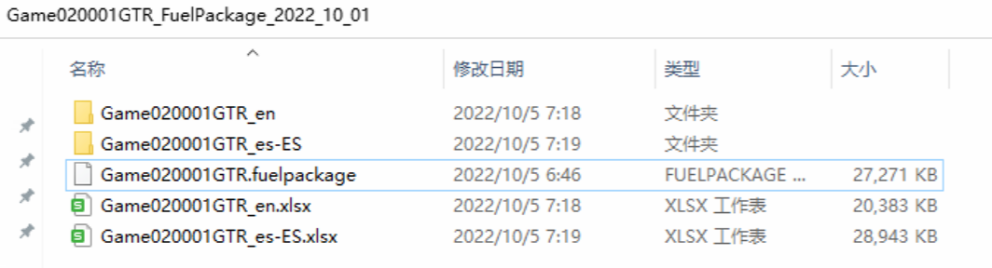




第四步，运行FUEL Translation Adapter工具软件，切换到Local Package选项卡，将“.fuelpackage”文件拖拽进去，然后勾选English和Spanish选项，最后点击Convert to Excel按钮，将翻译内容导出到“.fuelpackage”文件所在文件夹内。



导出成功后内容应该如下所示：



最后，将该文件夹直接打包成Zip文件（不允许使用其他压缩格式），名称和文件夹保持一致，内部应该只有这一个文件夹，不包含中文，并且没有任何嵌套，如下所示：





* **IGT打包UCGS**

方式和主工程类似，UCGS只需要打源码包。

首先，将Library和Obj文件夹移到工程以外，防止被包含。



然后，将projects、PRV、“.cfg”文件、“.bat”四个部分直接压缩成Zip文件（不允许使用其他压缩格式），命名格式为“项目编号\_类型\_日期”，同样不允许有任何嵌套或不相关文件出现，这几个文件的作用和主工程的一样，PRV文件夹用于存放Defect修改记录，前期没有可以不包含，最终结果如下所示：



1. **版本修复记录的格式及递交说明**

版本名称：XXX01\_OSX\_XXXX\_XX\_XX

更新描述：这里需要描述清楚和上一个版本的不同之处，多数情况下应该写修复了那些Bug和更新了那些内容，第一次递交的时候应该写清楚是完成了逻辑版本还是完整版本（包含所有美术素材）。

待修复：这个如果没有可以不用填写，如果有就需要写清楚。

* **修复记录的格式**

新建一份Word文档，命名为“XXXXX\_ChangeList”,每次要上传的版本都需要参照以上格式补充进文档，中间以横线隔开，横线上下各空一行，保持格式工整，如下所示：

版本名称：XXX01\_OSX\_2022\_10\_01

更新描述：XXXXXXXXXXXXXXX

待修复：XXXXXXXXXX

版本名称：XXX01\_OSX\_2022\_10\_02

更新描述：XXXXXXXXXXXXXXX

待修复：XXXXXXXXXX

**IGT安装的版本在正式递交PA之前以及其他所有的项目都应该参照这份格式进行相关问题记录！！！**

* **版本递交说明**

Demo以及公司内部的其他项目每次上传新的版本必须包含一份ChangeList，目的是让负责人明确更新内容，有的时候可能是将版本上传到一些第三方平台，这个时候需要将最新的修复记录发送至Skype相关项目组群里，确保负责人知晓。

IGT安装的项目在内测期间，若需要上传给公司内部负责人Review同样需要按照该方式进行，如果是需要递交给客户Review则不需要包含changeList。

每个版本在上传成功后，需要在Skype相关项目组群里@相关负责人并说明项目上传到了哪里（公司云用Cloud，IGT的云用SharePoint，第三方平台就说明平台名称）例如：

XXX01最新版本已上传至Cloud\Review\OCT\_01文件夹内

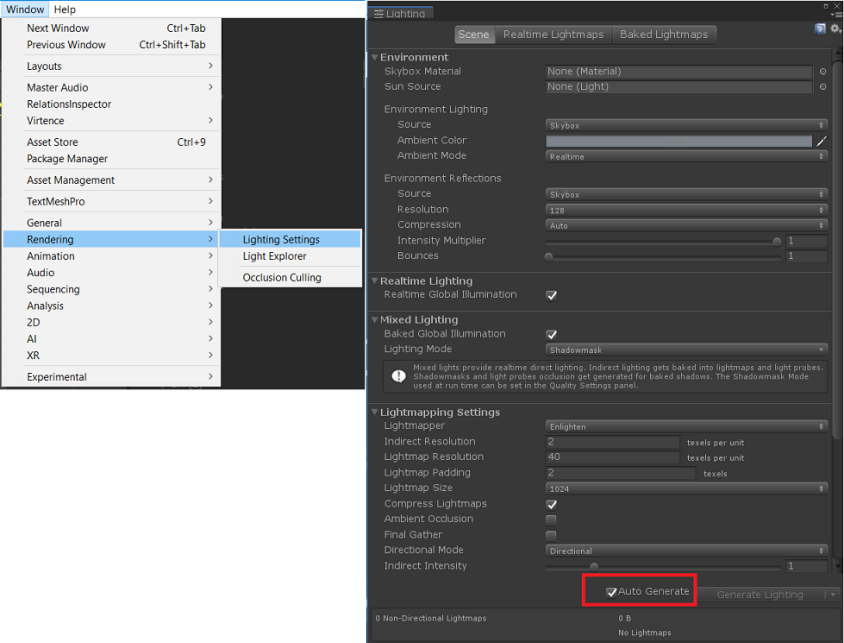
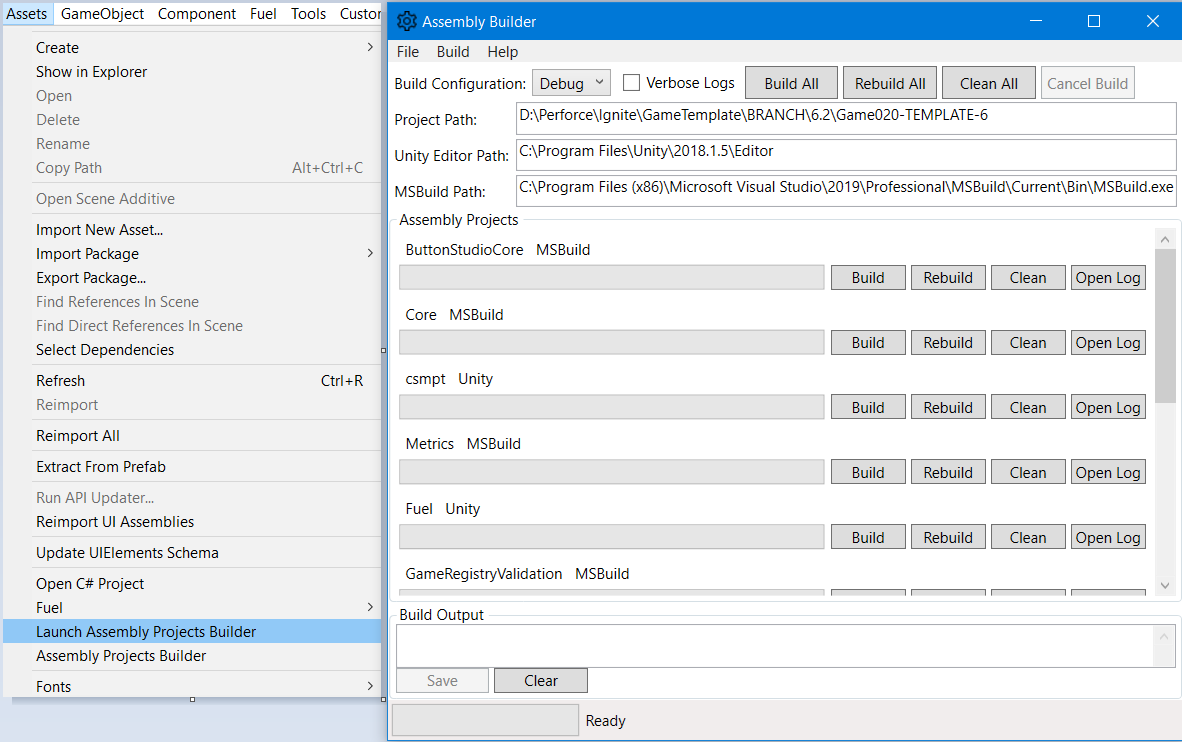
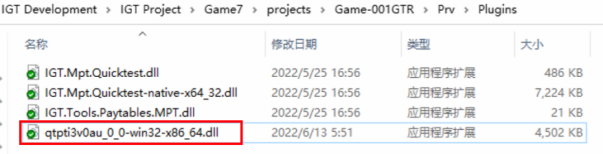
IGT01 最新版本已上传至SharePoint\From Rising Digital\IGT01\Integration文件夹内

XXX项目最新版本已上传至TestFlight

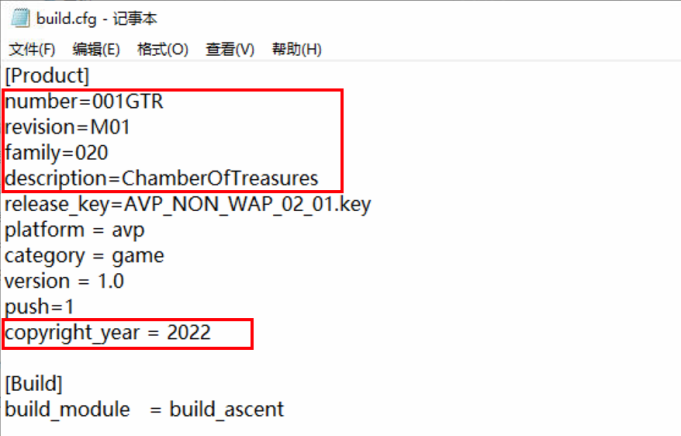
1. **版本正式递交内部审核规范**

* **Demo及公司内部项目**

项目的最终版本需要做好备份，源码需要在Git上同步至最新并标注好日期，确保日后开发使用不会出现丢失或版本错乱问题，这一类项目只需要保证负责人知晓即可。

* **IGT递交PA Test之前**
  + 需要完成游戏各个功能模块的开发，主要包括：游戏整体的玩法逻辑和画面表现、GameRule界面、完整的MPT、完整Paytable和Standalone+Link模式注册文件（这部分内容可参考“[如何创建PSMD注册文件](#如何创建PSMD文件)”部分，但是由于更新问题，可能有部分内容已经过时，所以最好的办法是使用之前游戏的注册文件直接修改）、内容翻译。
  + 需要按照游戏测试文档完成测试，确保无明显红屏、画面跳帧、导致游戏无法继续运行的逻辑错误等问题。
  + 做好内存分析和性能优化，确保没有内存泄漏和画面卡顿现象，游戏在EGM上运行的实时帧数最低不低于26帧（一般来说只要不卡就行）。
  + 整个游戏工程打包后的Zip包不超过3GB大小。
  + 确保游戏内所有场景关闭了自动生成光照贴图的功能（注意：需要为每个场景检查该设置）。****
  + 确保AssemblyProjects文件夹内的所有“.csproj”和“.sln”都被包含，可以运行AssemblyBuilder工具进行一次Rebuild，检查看看有无报错。
  + 确保Assets同级目录中的Prv\Plugins文件夹内只包含一个“qtpti”开头的“.dll”文件，具体用哪一个取决于游戏使用的MPT版本，记得确认。
  + 确保游戏工程内所有配置表、注册文件填写的名称或编号正确，主要包括以下几部分内容：

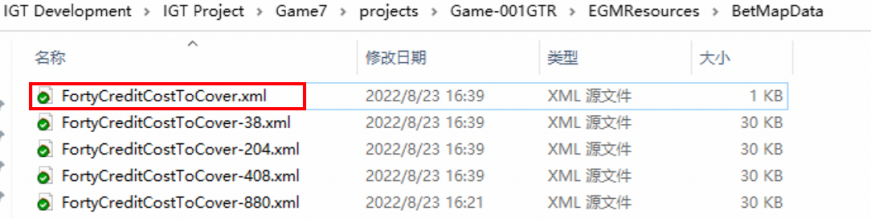
游戏根目录的“.cfg”文件和“.bat”文件的名字

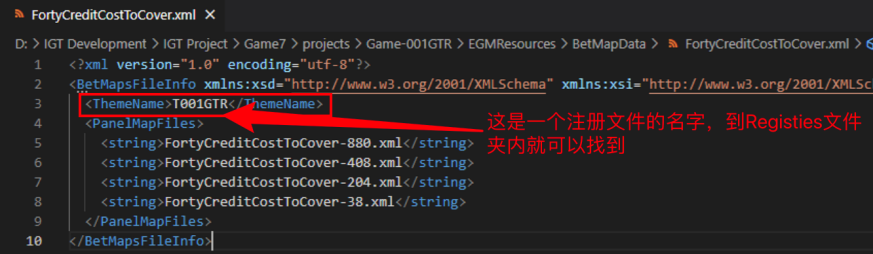
 

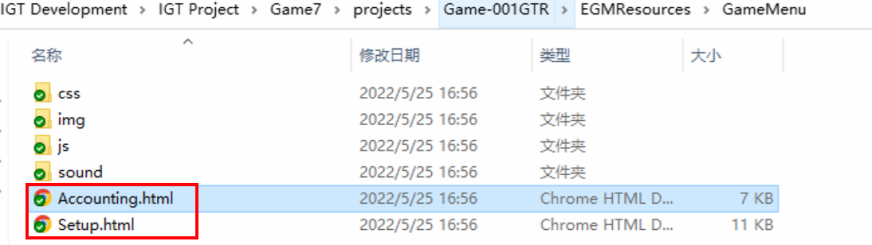
注意Projects文件夹内第一级文件夹名字的格式，用“-”代替了020

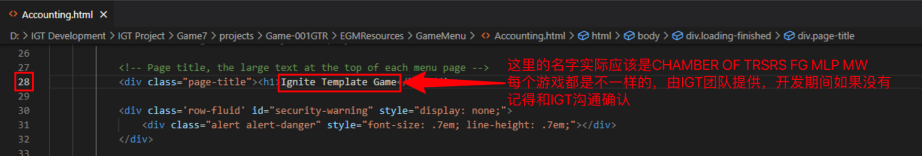


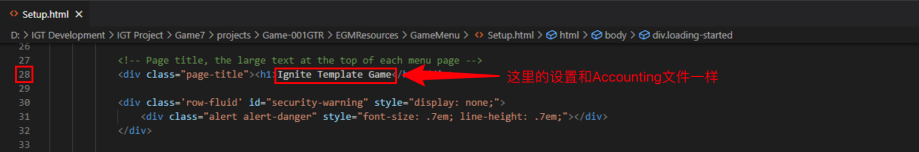
EGMResource目录下的BetMapData、GameMenu、LocalizedThemeContent文件夹内有一部分文件需要修改，见截图：

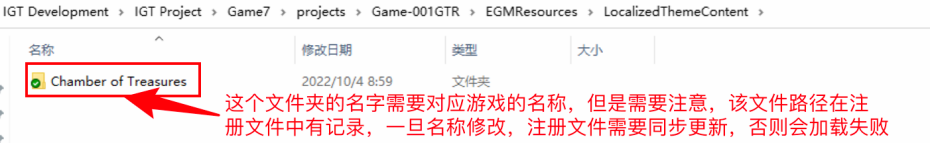


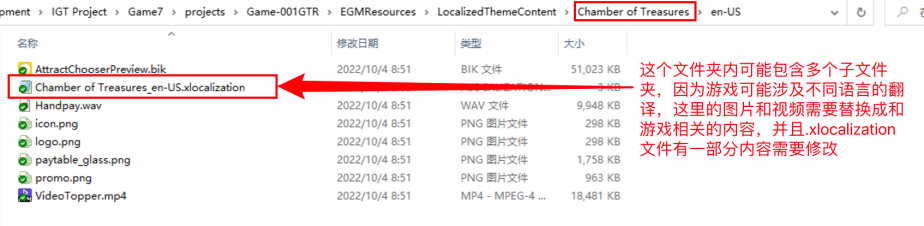






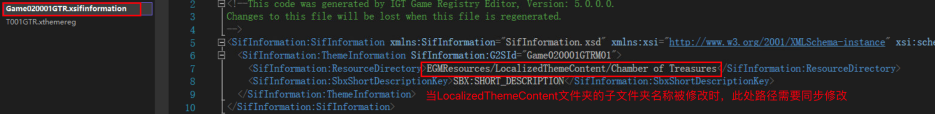
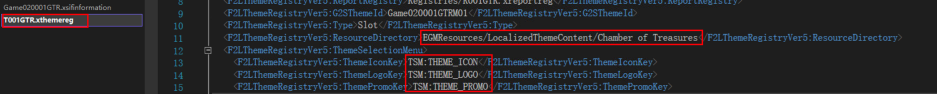






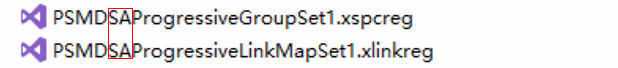


以下是调整过EGMResource目录中的内容后，可能需要调整的注册文件

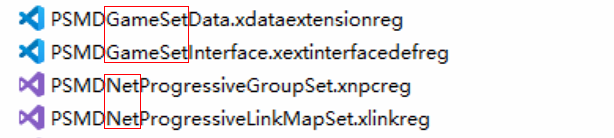
 

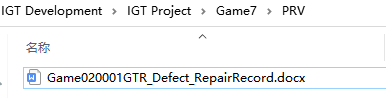
* **IGT正式递交PA**
  + Fuel和UCGS在打包完成之后可以直接上传至SharePoint，不需要其他操作。
  + IGT主工程源码包和EGM包在上传时需要替换注册文件（注意：经过和IGT确认，我们PA Test的版本不需要递交EGM包，只需要一个源码包，若有需要IGT会提要求）。游戏主工程的注册文件分为Standalone模式和Link模式两种，我们在制作注册文件时需要全部做好，我们本地测试用Standalone，**PA递交用Link**。

Standalone模式的文件我们通常以SA开头，分别以“.xspcreg”和“.xlinkreg”后缀作为结尾，仅包括以下两个文件：



Link模式的文件我们通常以Net开头(在最新要求中，Link模式可能需要很多套，每套对应一组Denomination和RTP的组合设置，若有多套需要全部包含)，分别以“.xnpcreg”和“.xlinkreg”后缀作为结尾，它还需要包括两个GameSet开头的文件（注意：这两个文件需要通过Jira先交给IGT加上数字签名，否则PA团队可能无法使用），分别以“.xdataxtensionreg”和“.xextinterfacedefreg”后缀作为结尾，参考以下文件：



* IGT主工程源码在递交之前，需要从压缩包在本地解压一份，目的有两个，第一个目的是检验压缩包是否有损坏，防止压缩包自身问题导致build失败，另一目的是需要运行“GameCheck”检测工具，检查包内文件是否包含不符合规范的问题（注意：该工具检测出的任何红色报错都必须修复，否则上传后必然出问题，在不知道如何修复的情况下，宁可不上传！务必确认该问题已经修复,紧急情况可以考虑将报错部分内容暂时还原）。
* 打包一份EGM包在本地测试机上运行一次试试，主要目的是为了防止万一有注册文件修改不正确会导致红屏，有些问题在Unity里面是没办法看到的，所以务必确认一次，若时间允许，应该仔细检查是否有其他问题。
* 若以上问题都顺利通过，就可以打开SharePoint上传源码包了，在上传完成后，需要通知相关负责人，具体方式见“[版本递交说明](#版本递交说明)”章节，包括后续每个版本更新。
* **IGT递交PA Test之后**
  + 正常情况下项目会在递交后一周之内开始测试，前期问题相对比较集中，我们一般会选择两到三周的时间作为一个版本周期（具体时间需要根据项目紧张程度来决定）。
  + PA 测试出来的问题通过Jira反馈（<https://valleyoffire.jira.igt.com>）,需要有IGT账号才能登录，测试期间要养成习惯，每天早上先检查一下Defect是否有更新，每个问题都附带一个编号，可用于Defect修复记录，每次递交需要将该版本修复的内容补充到Defect Change List里面，具体方式如下：
    - 先在项目根目录里新建一个名为“PRV”的文件夹，然后在里面新建一份以“[Product Name]\_Defect\_RepairRecord”格式命名的Word文档。
    - 在文档中首先记录每个版本的修复周期，然后按照每个Defect编号记录下修改说明以及修改了那些文件（注意：修改的文件如果是脚本类型，必须标记出修改了哪一行），格式如下：

-----------------------------------------------------

2022.10.01 —— 2022.10.12

-----------------------------------------------------

Defect 1-01：

Fill in the problem repair instructions here.

Affect files:

projects/XXXX/XXXX/XXXXX/XXXX.cs #63,66,84

projects/XXXX/XXXX/XXXXX/XXXX.cs #101-107

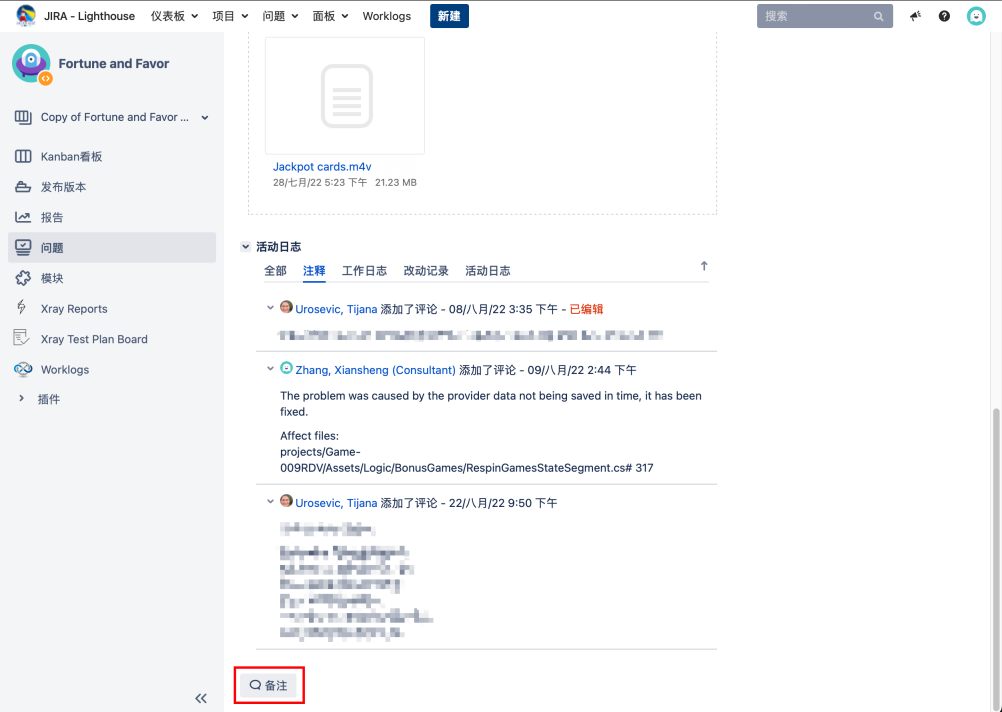
Defect 1-02：

Fill in the problem repair instructions here.

Affect files:

projects/XXXX/XXXX/XXXXX/XXXX.xpayvar

projects/XXXX/XXXX/XXXXX/XXXX.xml

* + PA Test版本递交过之后，每次再更新版本时就需要包含PRV文件夹了，具体方式见“[IGT打包主工程源码包](#IGT打包主工程源码)”章节。另外，每个问题的修复记录还需要在Jira上进行备注说明，每个Defect页面的最下方可以看到该选项，并进行答复，如下所示：

# 附件：

