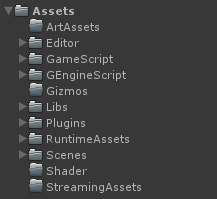
**程序目录及规范**

|  |  |  |
| --- | --- | --- |
| **日期** | **修改人** | **描述** |
| 2016-08-05 | 钟国良 | 添加文档草案 |

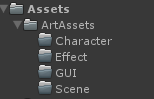
# Unity工程主目录



这里我们只关注项目开发中的各部门交互目录，常规Unity内置目录暂不在本文档内进行说明。

## 美术开发资源目录ArtAssets

ArtAssets目录用于存放美术同事开发的资源



目前主要分为四类美术资源：

* Character目录，存于角色模型资源，比如fbx和相应的Prefab
* Effect目录，存放特效资源，主要指粒子特效资源
* GUI目录，存放界面GUI资源，比如界面开发的原始贴图、动画、打包整合贴图等
* Scene目录，存放3D场景相关资源，场景物件、地形等

## 运行时资源目录RuntimeAssets

RuntimeAsset用于存放游戏运行时的非美术类资源，比如数值表的导出文件，程序制作的部分运行Prefab等。比如：



实际的目录，将在项目中再扩充

## 第三方扩展Libs

Libs目录主要用于存放开发时所需第三方插件、程序库等资源

## Shader目录

统一的Shader管理目录

## 程序脚本目录GameScript \GEngineScript

程序脚本主要分为C#代码和Lua代码，在代码文件的管理，采用模块分类，具体目录内容依据实际模块动态添加。主要说明：

* GameScript/Lua 目录，存在具体游戏逻辑模块的Lua代码
* GEngineScript/Lua 目录，存在游戏通用引擎模块Lua代码

## 特殊命名说明

公共资源目录使用“\_gy”结尾，以场景目录为例，公共物件目录SceneObject\_gy

# 关于美术资源命名建议

如果美术组的有更好的命名和资源管理方式，可以同技术商定，这里只是提供建议方案！

## 为什么要资源命名规则

* 防止文件重名，确保资源的唯一性。以防技术组在打包发布包体，造成资源丢失等不确定因素
* 方便资源查找管理

## 资源命名规则

资源命名规则，采用“前缀+资源名+序号“的组合方式。

### 关于前缀

|  |  |  |
| --- | --- | --- |
| **前缀** | **描述** | **示例** |
| Char\_ | 角色模型 | Char\_Role\_01.fbx |
| Eff\_ | 特效资源 | Eff\_Skill\_001.prefab |
| Sc\_ | 场景 | Sc\_mainCity\_01.unity |
| Gui\_ | GUI | Gui\_Login.prefab |